

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова праця
на правах рукопису

ГАЛУШКО ДМИТРО ОЛЕКСАНДРОВИЧ

УДК 004.05

ДИСЕРТАЦІЯ
УПРАВЛІННЯ ЯКІСТЮ ПОСЛУГ У КОРПОРАТИВНІЙ ІТ-ІНФРАСТРУКТУРІ

Спеціальність 05.13.06 – Інформаційні технології

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів
мають посилання на відповідне джерело

_____ Д.О. Галушко

Науковий керівник

Ролік Олександр Іванович
доктор технічних наук, професор

Київ – 2021

АНОТАЦІЯ

Галушко Д.О. Управління якістю послуг у корпоративній ІТ-інфраструктурі. На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. Робота виконана на кафедрі автоматики та управління в технічних системах Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Міністерства освіти і науки України. Захист відбудеться у спеціалізованій вченій раді Д 26.002.29 при Національному технічному університеті України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, 2021.

Дисертацію присвячено розробці комплексу математичних моделей, методів, алгоритмів та інформаційних технологій для визначення та управління рівнем якості ІТ-послуг, які надаються корпоративними ІТ-інфраструктурами.

Для впровадження та використання інформаційних технологій корпорації створюють власні ІТ-інфраструктури або використовують ІТ-інфраструктури, які надаються провайдерами хмарних технологій. ІТ-послуги, які надаються ІТ-інфраструктурами, можуть використовуватися корпораціями для власних потреб або надаватися стороннім організаціям та фізичним особам. В останньому випадку компанія набуває статусу провайдера ІТ-послуг.

Від ефективності функціонування ІТ-інфраструктури безпосередньо залежить якість ІТ-послуг і, як наслідок, ефективність ведення бізнесу в цілому. У тому випадку, коли ІТ-послуги, що надаються корпоративною ІТ-інфраструктурою, використовуються для автоматизації бізнес-процесів, які виконуються співробітниками корпорації, зниження рівня якості ІТ-послуг збільшує час виконання бізнес-операцій або взагалі унеможлиблює виконання деяких бізнес-процесів. Зниження, навіть короточасне, рівня якості надання ІТ-послуг провайдером цих послуг суттєво впливає на конкурентоспроможність та добутки провайдерів, оскільки несвоєчасне або неякісне надання ІТ-послуг може призвести до суттєвих економічних втрат користувачами ІТ-послуг. Тому власники ІТ-інфраструктур прикладають багато зусиль та витрачають багато

коштів на контроль якості ІТ-послуг і підтримку якості надання ІТ-послуг на узгодженому рівні.

Для управління ІТ-інфраструктурою з метою підтримки якості ІТ-послуг на узгодженому рівні, за умов раціонального використання інформаційно-обчислювальних ресурсів, розробляються та впроваджуються системи управління ІТ-інфраструктурою. Безперервне збільшення кількості використовуваних в компаніях інформаційних технологій та програм, суттєва складність ІТ-інфраструктур, яка постійно зростає, використання технологій віртуалізації і парадигми хмарних обчислень роблять завдання управління ІТ-інфраструктурою далеко не тривіальним. При оперативному автоматичному управлінні ІТ-інфраструктурою з ціллю підтримки якості надання ІТ-послуг на узгодженому рівні в системи управління ІТ-інфраструктурі необхідно вирішувати завдання оцінки поточного стану надання ІТ-послуг, оцінки фактичного рівня якості цих послуг, визначення та вироблення керуючого впливу на компоненти ІТ-інфраструктури задля відновлення або підтримки якості надання ІТ-послуг на узгодженому рівні.

Компанії NetCracker, Oracle, CISCO, Microsoft, IBM, Intel та ін. займаються розробкою систем управління ІТ-інфраструктурами, які здатні автоматично підтримувати працездатність та рівень якості послуг відповідно до різноманітних сценаріїв виходу з ладу окремих компонентів ІТ-інфраструктури, яка надає ІТ-послуги компаніям та окремим користувачам. Розробкою методів управління ІТ-інфраструктурами займаються спеціалісти з усього світу: науковці КПІ ім. Ігоря Сікорського ведуть активні наукові дослідження з управління ІТ-інфраструктурами на основі сервісно-орієнтованого та декомпозиційно-компенсаційного підходів; професори та аспіранти зі шведського університету “Umea University” та хорватського університету “University of Split” вивчають вплив якості та вартості надаваних сервісів на рівень задоволеності клієнтів, а також розробляють рекомендації щодо нормативних значень параметрів якості ІТ-послуг; корейським університетом Kwangwoon University розробляються алгоритми та методи, що забезпечують надання більш якісних ІТ-послуг. Не зважаючи на те, що існують технології, які дозволяють системам управління ІТ-

інфраструктурою автоматично реагувати на вихід з ладу обчислювального та комунікаційного обладнання або інших компонентів ІТ-інфраструктури, комплексного рішення, яке на основі даних моніторингу засобів надання ІТ-послуг та оцінки показників рівня якості послуг має можливість знаходити причини падіння рівня якості сервісу досі немає.

Використання запропонованих в роботі методів та моделей управління якістю послуг у корпоративних ІТ-інфраструктурах, створення на їх основі інформаційних технологій управління якістю ІТ-послуг в умовах консолідації інформаційно-обчислювальних ресурсів і використання засобів віртуалізації, та подальше використання цих ІТ у різноманітних системах управління ІТ-інфраструктурами корпоративного рівня для автоматичного управління рівнем якості ІТ-сервісів, дозволить не лише підтримувати узгоджений рівень ІТ-послуг, а ще й раціонально використовувати інформаційно-обчислювальні ресурси ІТ-інфраструктури, які задіюються для надання цих послуг. Тому дана робота, яка присвячена управлінню корпоративними ІТ-інфраструктурами задля забезпечення та підтримки заданого рівня якості ІТ-послуг, є актуальною.

Об'єктами дослідження є ІТ-послуги, які надаються користувачам корпоративної ІТ-інфраструктури.

Предмет дослідження – методи та засоби управління рівнем якості ІТ-послуг, які забезпечують підтримку узгодженого рівня якості ІТ-послуг за умов раціонального використання ресурсів ІТ-інфраструктури корпоративного рівня.

Метою роботи є забезпечення підтримки якості ІТ-послуг на заданому рівні шляхом розроблення та впровадження моделей, методів, інформаційних технологій та інструментів керування ІТ-інфраструктурою за умов раціонального використання її консолідованих інформаційно-обчислювальних ресурсів.

Результати дослідження викладено у чотирьох розділах дисертації.

Перший розділ присвячено огляду та аналізу проблем управління якістю ІТ-послуг та постановці задачі дослідження. Вводяться основні поняття та визначення ІТ-сервісів та ІТ-послуг. Досліджується проблема аналізу якості ІТ-послуг, наводяться рівні аналізу та критерії якості ІТ-послуг. Здійснюється огляд

типів віртуалізації та їх використання у ІТ-інфраструктурах, що надають ІТ-послуги. Проаналізована задача визначення рівня задоволеності користувача якістю сервісів, що надаються яка зводиться до задачі визначення залежності якості надаваних сервісів від значень окремих параметрів якості ІТ-послуг. Досліджено проблему створення системи автоматичного керування рівнем якості ІТ-послуг. Уточнені задачі дослідження.

Другий розділ присвячено розробці моделей і алгоритмів оцінки рівня якості ІТ-послуг. Запропоновано моделі та методи оцінки рівня якості ІТ-послуг, що надаються провайдером послуг. Запропоновано метод зведення рівнів якості декількох ІТ-послуг до спільного інтегрального показника якості роботи ІТ-сервісів.

Третій розділ присвячено моделей, методів і алгоритмів управління ресурсами ІТ-інфраструктури, що надає ІТ-послуги, для забезпечення функціонування послуг на узгодженому з замовником рівні. Розроблено методи розподілу ресурсів ІТ-інфраструктури провайдеру сервісів для забезпечення узгодженого рівня якості ІТ-послуг при використанні мінімального об'єму ресурсів ІТ-інфраструктури.

В четвертому розділі розглядаються технологічні аспекти реалізації інформаційної технології управління рівнем якості ІТ-послуг та перерозподілу ресурсів ІТ-інфраструктури для підтримки якості послуг на узгодженому рівні. Розглянуті у попередніх розділах, моделі та алгоритми становлять складові теоретичних основ проектуванні інформаційної технології управління рівнем якості ІТ-послуг та ресурсами ІТ-інфраструктури. Технологію, розроблену на базі запропонованих в роботі алгоритмів, моделей та методів встановлено на одному з хостингів веб сторінок та проведено досліди, що підтвердили її ефективність.

Ключові слова: нечітка логіка, нейронна мережа, системи автоматичного управління, якість сервісів, якість послуг, інформаційні технології, ІТ-сервіси, ІТ-послуги, ІТ-інфраструктура.

ABSTRACT

Dmytro Halushko, Quality management of services in the corporate IT infrastructure. On the rights of the manuscript.

The dissertation on competition of a scientific degree of the candidate of technical sciences on a specialty 05.13.06 – Information Technology. The work was performed at the Department of Automation and Control in Technical Systems of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". The dissertation will be held at the specialized scientific council D 26.002.29 of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, 2021.

The dissertation is devoted to developing of mathematical models, methods, algorithms and information technologies for quality managing of IT services that are provided by corporate IT infrastructures.

Corporations create their own IT infrastructures or use IT infrastructures provided by cloud technology providers to implement and use information technology. IT services provided by IT infrastructures can be used by corporations for their own needs or can be provided to the third parties. In the last case, the company acquires the status of an IT service provider.

The efficiency of IT infrastructure directly affects the quality of IT services and, consequently, the efficiency of doing business in general. When IT services provided by the corporate IT infrastructure are used to automate business processes performed by employees of the corporation, reducing the quality of IT services increases the time of business operations or makes it impossible to perform some business processes. The reduction, even in the short term, of the quality of IT services provided by an IT service provider significantly affects the competitiveness and performance of providers, as untimely or poor quality IT service provision can lead to significant economic losses for IT service provider customers. Therefore, IT infrastructure owners have to make a lot of effort and money to control the quality of IT services, maintain the quality of IT services at an agreed level and respond in a timely manner to events that lead or may lead to a decline in the quality of IT services.

To manage the IT infrastructure in order to maintain the quality of IT services at an agreed level, under the conditions of rational use of information and computing resources, IT infrastructure management systems are developed and implemented. The continuous increase in the number of information technologies and programs used in companies, the significant complexity of IT infrastructures, which is constantly growing, the use of virtualization technologies and the paradigm of cloud computing make the task of IT infrastructure management far from trivial. In the operational automatic management of IT infrastructure in order to maintain the quality of IT services at an agreed level in IT infrastructure management systems, it is necessary to solve the problem of assessing the current state of IT services, assessing the actual level of service quality, identifying and developing management influence on IT components. infrastructure to restore or maintain the quality of IT services at an agreed level.

NetCracker, Oracle, CISCO, Microsoft, IBM, Intel and others. are engaged in the development of IT service management systems that allow you to automatically maintain the performance of services in accordance with various scenarios of failure of individual components of the IT infrastructure, I provide IT services to companies and individual users. The solution to this problem, as well as the development of methods for managing IT infrastructures are specialists from around the world: scientists KPI. Igor Sikorsky conducts active research on IT infrastructure management on the basis of service-oriented and decomposition-compensation approaches; professors and graduate students from the Swedish University "Umea University" and the Croatian University "University of Split" study the impact of quality and cost of services on customer satisfaction, as well as develop recommendations on regulatory values of quality parameters of IT services; Kwangwoon University in Korea is developing algorithms and methods to provide better IT services. Although there are technologies that allow IT infrastructure management systems to automatically respond to the failure of computer and communication equipment or other components of the IT infrastructure, a comprehensive solution that is based on data from the monitoring of IT services and assessment of performance indicators quality of IT

services has the ability to find the reasons for the decline in the level of service quality is still not.

Use of the methods and models of quality management of services in corporate IT infrastructures offered in work, creation on their basis of information technologies of management of quality of IT services in the conditions of consolidation of information and computer resources and use of means of virtualization, and further use of these IT in various IT management systems. corporate-level infrastructures for automatic quality management of IT services will allow not only to maintain a consistent level of IT services, but also to rationally use information and computing resources that are used to provide these services. Therefore, this work, which is devoted to the management of corporate IT infrastructures to ensure and maintain a given level of quality of IT services, is relevant.

The objects of research are IT services provided to users of corporate IT infrastructure.

The subject of research - methods and tools for managing the level of quality of IT services, which provide support for a consistent level of quality of IT services in terms of rational use of IT infrastructure resources at the corporate level.

The aim of the work is to ensure the maintenance of the quality of IT services at a given level by developing and implementing models, methods, information technologies and tools for managing IT infrastructure under the conditions of rational use of its consolidated information and computing resources.

The results of the research are presented in four sections of the dissertation.

The first section is devoted to reviewing and analyzing the problems of quality management of IT services and setting the research task. The basic concepts and definitions of IT services and IT services are introduced. The problem of quality analysis of IT services is investigated, levels of analysis and criteria of quality of IT services are given. An overview of the types of virtualization and their use in IT infrastructures that provide IT services. The problem of determining the level of user satisfaction with the quality of services provided is reduced to the problem of determining the dependence of the quality of services provided on the values of individual parameters of the quality of IT services. The problem of creating a system

of automatic quality management of IT services is investigated. The research tasks are specified.

The second section is devoted to the development of models and algorithms for assessing the level of quality of IT services. Models and methods for assessing the level of quality of IT services provided by the service provider are proposed. A method of reducing the quality levels of several services to a common integrated indicator of the quality of services is proposed.

The third section is devoted to models, methods and algorithms for managing the resources of the IT infrastructure that provides IT services to ensure the functioning of services at the level agreed with the customer. Methods for allocating IT infrastructure resources to a service provider have been developed to ensure a consistent level of IT service quality while using a minimum amount of IT infrastructure resources.

The fourth section discusses the technological aspects of the implementation of information technology for managing the level of quality of IT services and redistribution of resources of the IT infrastructure of the provider that provides these services. The models and algorithms considered in the previous sections are components of the theoretical basis for designing information technology for managing the level of quality of IT services and IT infrastructure resources. The technology developed on the basis of the algorithms, models and methods described in the work was installed on one of the web page hosts and experiments were performed that confirmed its effectiveness.

Keywords: fuzzy logic, neural network, automatic control systems, quality of services, quality of services, information technologies, IT services, IT infrastructure.

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЙНОЇ РОБОТИ / THE LIST OF AUTHOR'S PUBLICATIONS FOR DISSERTATION THESES

Безпосередньо за тематикою дисертаційної роботи опубліковано 18 наукових робіт, а саме: / Directly for the subject of dissertation theses 18 scientific publications have been published:

Наукові публікації, в яких висвітлені основні наукові результати дисертації (18) / The main scientific results of the dissertation theses were published in the next publications (18):

– статті в наукових фахових журналах України (3) / articles in Ukrainian professional journals (3):

1. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: ВЕК+, 2010. – № 52. – С. 39–52

2. Ролик А.И. Оценка качества предоставления мультимедийных сервисов с использованием нейросетевого классификатора / Ролик А.И., Галушко Д.А., Барна В.В., Томащук А.В., Ясочка М.В. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Століття +, 2015. – № 63. – С. 25–30.

3. Ролик А.И. Метод нечеткой непараметрической оценки качества функционирования элементов информационно-телекоммуникационных систем / А.И. Ролик, Д.А. Галушко, П.Ф. Можаровский, А.А. Андриенко, В.М. Вовк // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2012. – Вип. 19 (39). – С. 115–125.

– статті в закордонних фахових виданнях, які реферуються базою Scopus (1), в тому числі: / articles in foreign professional journals referenced by the Scopus database (1), including:

– третього квартиля (Q3) (1) / the third quartile (Q3) (1):

4. Rolik O., Kolesnik V., Halushko D. Decomposition-Compensation Method for IT Service Management. In: Kulczycki P., Kóczy L., Mesiar R., Kacprzyk J. (eds) Information Technology and Computational Physics. CITCEP 2016. Advances in Intelligent Systems and Computing, vol 462, Springer, Cham, 2017, pp. 89–107.

– За результатами досліджень також опубліковано статті в інших наукових журналах України та за кордоном (2) / According to the results of investigations, the articles in other scientific Ukrainian and foreign journals are also published (2):

5. Telenyk S. Models and methods of resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko // Technical transaction. Automatic control. – Politechnica Krakowska, 2013. – vol. 4-AC. – pp. 41–52.

6. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // Information Technology, Computational and Experimental Physics. Kulczycki P., Kowalski P. A., Łukasik S. (eds). – AGN-UST. – 2016. – pp. 13–16.

Результати, викладені в дисертації, було апробовано на 12 науковій конференції, зокрема опубліковано 12 матеріалів конференції / The results shown in the theses were validated at 12 scientific conferences, in particular, were published in 12 materials of the conferences:

- що реферується базою Scopus (3) / which are referenced by the Scopus database (3):

7. Rolik O. Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm / O. Rolik, V. Kolesnik and D. Halushko// Proc. of the International Conference Radio Electronics & Inf

8. Qualitative evaluation method of IT-infrastructure elements functioning / S. Telenyk, O. Rolick, M. Bukasov, Y. Dorogiy, D. Halushko, A. Pysarenko // : 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Odessa, 2014, pp. 165–169.

9. Qualitative evaluation method of IT-infrastructure functioning based on structural optimization of neural network // Y. Dorogyi, O. Doroga-Ivaniuk, S. Telenik and D. Halushko // 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, Ukraine, 2015, pp. 1–4.

- інші конференції (9) / other conferences (9):

10. Теленик С.Ф. Система управління ІТ-інфраструктурой – путь к повышению эффективности функционирования предприятия / С.Ф. Теленик, А.И. Ролик, М.М. Букасов, А.В. Волошин, Д.А. Галушко // Інформаційні технології – інструмент підвищення конкурентоздатності підприємств: 9–10 груд. 2009.: матеріали конф. – К.: УкрНЦ РІТ, 2009. – С. 30–33.

11. Ролик А.И. Метод оценки состояния элементов информационно-телекоммуникационных систем на основе нечетких нейронных сетей/ А.И. Ролик, Д.А. Галушко, Ю.А. Кононенко // Обчислювальний інтелект (Результати, проблеми, перспективи): матеріали II-ї Міжнар. наук.-техн. конф. «Обчислювальний інтелект (QI-2013)», м. Черкаси, 14-18 травня 2013 р. – Черкаси: – 2013. – С. 233

12. Ролік О.І. Метод оцінки стану елементів інформаційно-телекомунікаційних систем на базі апарату нечіткої логіки / О.І. Ролік, Д.О. Галушко, І.М. Плехова, Б.В. Ступак // : матеріали конф. Physical and technological problems of radio engineering devices, telecommunication, nano- and microelectronics. Proceedings of the 3th international scientific-practical conference. Chernivtsi, Ukraine, October 24–26, 2013. – Чернівці: КОД. – 2013. – С. 104–105.

13. Ролик А.И. Метод оценки качества телекоммуникационных сервисов на основе нейронных сетей со структурным обучением / А.И. Ролик, Д.А. Галушко, Д.С. Захаров, А.В. Томащук// Обчислювальний інтелект (результати, проблеми, перспективи): III-я Міжнародна наук.-практ. конф. 12-15 травня 2015 р. Київ-Черкаси: матеріали. – Черкаси: видавець Чабаненко Ю., 2015. – С. 250–251.

14. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // : матеріали

конф. Proc. of the Congress on Information Technology, Computational and Experimental Physics 2013 (CITCEP'15) 18–20 December, Cracow, Poland. – 2015. – p. 176–179.

15. Ролик А.И. Управление ресурсами виртуальной ИТ-инфраструктуры для поддержания заданного уровня предоставляемых сервисов / А.И. Ролик, Д.А. Галушко, Н.В. Кравчун, Т.В. Кравченко // : матеріали конф. ICACIT-2015: 3-я Міжнар. конф. з автоматичного управління та інформаційних технологій, 11-13 грудня 2015 р. м. Київ: матеріали. – К., 2015. – с. 80–83.

16. Rolik O. Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm / O. Rolik, V. Kolesnik and D. Halushko// Proc. of the International Conference Radio Electronics & Info Communications (UkrMiCo). – Kiev, Ukraine. – 2016. – pp. 1–5.

17. D. Halushko, O. Rolik, and V. Samoty, “A Load Balancing Mechanism Based on Fuzzy Nonparametric Analysis of QoS Parameters” in Proc. CLOUD COMPUTING 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, February 19-23, 2017 Athens, Greece, IARIA, 2017, матеріали конф; pp. 102–107.

18. IT Service Quality Management Based on Fuzzy Logic O Rolik, V Kolesnik, D Halushko International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018 – Proceedings 31 January 2019, матеріали конф. pp. 604–608.

Всього за тематикою дисертації:

За тематикою дисертаційної роботи опубліковано 18 наукових робіт. Основні наукові результати висвітлені в 18 наукових публікаціях.

Зокрема, опубліковано 3 статті в наукових фахових журналах України; 1 наукова стаття у закордонних виданнях, проіндексованих у базі даних Scopus, з яких 1 стаття в журналах 3-го квартиля (Q3).

Крім того, результати роботи відображено 12 публікаціях у матеріалах науково-технічних конференцій, з яких 3 видань проіндексовано у базі даних Scopus.

Total for the dissertation theses:

On the topic of the dissertation, 18 scientific publications have been published. The main scientific results are shown in 18 scientific publications.

In particular, the following scientific publications have been published: 3 articles in the professional scientific journals of Ukraine; 1 articles in professional foreign journals indexed in the reference base Scopus, where 1 articles are in the journals of 3-rd quartile (Q3).

Besides, the results are presented in 12 proceedings of scientific conferences, including 3 proceedings indexed in the reference base Scopus.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	17
ВСТУП.....	18
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМ УПРАВЛІННЯ ЯКІСТЮ ІТ-ПОСЛУГ.....	27
1.1 Особливості ІТ-послуги як об'єкту управління.....	27
1.2 Аналіз проблем визначення якості ІТ-сервісів	33
1.3 ІТ-інфраструктури постачальника ІТ-сервісів	40
1.4 Особливості віртуалізації при управлінні ресурсами ІТ-інфраструктури	43
1.5 Узагальнена модель керування якістю ІТ-сервісів.....	48
РОЗДІЛ 2 МОДЕЛІ І МЕТОДИ ОЦІНЮВАННЯ ТА УПРАВЛІННЯ РІВНЕМ ЯКІСТІ ІТ-ПОСЛУГ	51
2.1 Підхід до управління рівнем якості надаваних ІТ-послуг	51
2.2 Метод оцінки якості ІТ- сервісів на основі простих нейромережових класифікаторів	58
2.2.1 Розробка методу оцінки якості ІТ-послуг на основі простих нейромережових класифікаторів	58
2.2.2 Вибір методу навчання штучної нейронної мережі	62
2.2.3 Метод оцінки якості роботи ІТ-сервісу	71
2.3 Структурна оптимізація нейромережового класифікатора.....	72
2.3.2 Основні структурні операції в мережі	74
2.4 Метод нечіткої непараметричної оцінки якості сервісу	84
2.4.1 Обґрунтування концептуальних положень запропонованого методу.....	84
2.4.2 Розрахунок рівня якості ІТ-сервісу	86
РОЗДІЛ 3 МОДЕЛІ І МЕТОДИ КЕРУВАННЯ ЯКІСТЮ ІТ-СЕРВІСУ.....	89
3.1 Розроблення узагальненої схеми управління ІТ-інфраструктурою.....	89
3.2 Управління ресурсами ІТ-інфраструктури, що надаються для функціонування ІТ-послуги	90
3.3 Управління ресурсами при балансуванні навантажень на елементи ІТ- інфраструктури.....	93
3.4 Керування якістю сервісу на основі апарату нечіткої логіки	97

3.5 Динамічний перерозподіл ресурсів	102
РОЗДІЛ 4 РОЗРОБЛЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ	
ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ ЯКІСТЮ	
КОРПОРАТИВНИХ ІТ-СЕРВІСІВ	105
4.1 Проектування програмного продукту для визначення якості	
мультимедійних сервісів	105
4.1.2 Шар доступу до даних	107
4.1.3 Бізнес шар	109
4.1.4 Шар наскрізної функціональності	112
4.1.5 Модуль внутрішньої взаємодії.....	112
4.2 Розроблення стенду для проведення дослідження середовища надання ІТ-	
послуг HTTP сервісу	113
4.2.2 Досліди з участю одного користувача	116
4.2.3 Досліди з великою кількістю користувачів	124
4.2.4 Оцінка якості функціонування сервісу після управління	131
ВИСНОВКИ.....	134
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	137

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

KVM – віртуальна машина ядра Linux

MTBF – Mean time between failures – Середній наробіток між відмовами

MTTR – Mean time to repair – Середній наробіток для ремонту

SLA – Service-level agreement – Угода про рівень послуг

QoS – Quality of service – Якість обслуговування

АНП – апарат нечіткої логіки

АС – асиметричний сервіс

БСММ – бізнес-сутність математичної моделі

ВМ – віртуальна машина

ІКТ – інформаційно-телекомунікаційні технології

ІТ – інформаційні технології

ОС – операційна система

ПЗ – програмне забезпечення

СНЧ – система, що не працює в режимі реального часу

СРЧ – система, що працює в режимі реального часу

СС – симетричний сервіс

СУІ – система управління інфраструктурою

ЧЗ – час завантаження

ШНМ – штучна нейронна мережа

ВСТУП

Актуальність теми. Сьогодні інформаційних технологій (ІТ) використовуються у всіх сферах людської діяльності. Повсюдне використання ІТ є характерною рисою ведення сучасного бізнесу. Це пояснюється тим, що завдяки використанню ІТ з'являється можливість автоматизувати майже усі бізнес-процеси. Автоматизація бізнес-процесів сприяє підвищенню ефективності ведення бізнесу в цілому.

Для впровадження та використання інформаційних технологій корпорації створюють ІТ-інфраструктури. ІТ-послуги, які надаються ІТ-інфраструктурами, можуть використовуватися корпораціями для власних потреб або надаватися стороннім організаціям та фізичним особам. В останньому випадку компанія набуває статусу провайдера ІТ-послуг.

Від ефективності функціонування ІТ-інфраструктури безпосередньо залежить якість ІТ-послуг і, як наслідок, ефективність ведення бізнесу в цілому. У тому випадку, коли ІТ-послуги, що надаються корпоративною ІТ-інфраструктурою, використовуються для автоматизації бізнес-процесів, які виконуються співробітниками корпорації, зниження рівня якості ІТ-послуг збільшує час виконання бізнес-операцій або взагалі унеможлиблює виконання деяких бізнес-процесів. Зниження, навіть короточасне, рівня якості надання ІТ-послуг провайдером ІТ-послуг суттєво впливає на конкурентоспроможність та добуток провайдерів, оскільки несвоєчасне або неякісне надання ІТ-послуг може призвести до суттєвих економічних втрат. Тому власники ІТ-інфраструктур змушені прикладати багато зусиль та витратити багато коштів на контроль якості ІТ-послуг, підтримку якості надання ІТ-послуг на узгодженому рівні та вчасно реагувати на події, що призводять, або можуть призвести до зниження рівня якості ІТ-послуг.

Для управління ІТ-інфраструктурою з метою підтримки якості ІТ-послуг на узгодженому рівні, за умов раціонального використання інформаційно-обчислювальних ресурсів, розробляються та впроваджуються системи управління ІТ-інфраструктурою (СУІ). Безперервне збільшення кількості

використовуваних в компаніях інформаційних технологій та програм, суттєва складність ІТ-інфраструктур, яка постійно зростає, використання технологій віртуалізації і парадигми хмарних обчислень роблять завдання управління ІТ-інфраструктурою далеко не тривіальним. При оперативному автоматичному управлінні ІТ-інфраструктурою з ціллю підтримки якості надання ІТ-послуг на узгодженому рівні в СУІ необхідно вирішувати завдання оцінки поточного стану надання ІТ-послуг, оцінки фактичного рівня якості послуг, визначення та вироблення керуючого впливу на компоненти ІТ-інфраструктури задля відновлення або підтримки якості надання ІТ-послуг на узгодженому рівні.

Сьогодні багато відомих компаній, серед яких NetCracker, Oracle, CISCO, Microsoft, IBM, Intel розробляють системи управління ІТ-послугами. Вони пропонують рішення для управління існуючими ІТ-інфраструктурами і створюють новітні принципи керування ІТ-інфраструктурами. Так, компанія Google створила відкрите програмне забезпечення Kubernetes, яке дозволяє керувати контейнерами Linux, що об'єднуються в систему віртуальних серверів та слугують постачальниками ІТ-послуг. Даним напрямом цікавляться і науковці з провідних світових університетів. У КПІ ім. Ігоря Сікорського ведуться активні наукові дослідження з управління ІТ-інфраструктурами на основі сервісно-орієнтованого та декомпозиційно-компенсаційного підходів; професори та аспіранти зі шведського університету “Umea University” та хорватського університету “University of Split” вивчають вплив якості та вартості надаваних сервісів на рівень задоволеності клієнтів, а також розробляють рекомендації щодо нормативних значень параметрів якості ІТ-сервісів; корейським університетом Kwangwoon University розробляються алгоритми та методи, що забезпечують надання більш якісних ІТ-послуг.

Не зважаючи на велику кількість компаній, що розробляють програмні продукти для управління ІТ-інфраструктурою, в їх основі, здебільшого, лежить алгоритми реакції на вихід з ладу обладнання або інших елементів ІТ-інфраструктури, або розглядаються алгоритми керування окремими конкретними сервісами, що надаються ІТ-інфраструктурою. Комплексного рішення, що базується не на якості функціонування елементів ІТ-інфраструктури

як постачальників сервісу, а на якості самих ІТ-послуг, що надаються інфраструктурою, не існує.

Компанії, що зацікавлені у вирішенні задач управління якістю ІТ-послуг, об'єднуються з науковцями та створюють міжнародні організації або форуми (TM Forum, Office of Government Commerce, IEEE, ISO та ін.), які займаються проблематикою управління якістю послуг та керування ІТ-інфраструктурою, чи розробляють власні рішення (IBM, Microsoft, Hewlett-Packard, Cisco, BMC та ін.). Управлінню ІТ-інфраструктурами присвячено ряд наукових робіт (Глушков В.М., Палагін О.В. та ін.). Управління ІТ-інфраструктурою спирається на такі напрями наукових досліджень як: системний аналіз (Вернадський В.І., Згуровський М.З., Панкратова Н.Д. та ін.), математичні методи вирішення задач оптимізації (Данціг Д.Б., Михалевич В.С., Павлов О.А., Кузнецов М.Ю., та ін.), інформаційні технології та системи (Перевозчикова О.Л., Сергієнко І.В., Андон П.І., Теленик С.Ф., Ролік О.І., Фіпсу Дж. та ін.), управління якістю послуг (Демінг У., Джуран Д., Шьюхард Д. та ін.), теорія управління багатооб'єктними системами (Кунцевич В.М., Лебедев Д.В., Подладчиков В.М., Кузнецов М.А., Месарович М., Беллман Р. та ін.), теорія штучного інтелекту (Хопфілд Д.Д., Голдберг Д., Зайченко Ю.П., Гуляницький Л.Ф., Бідюк П.І. та ін.).

Використання запропонованих в роботі методів та моделей управління якістю послуг у ІТ-інфраструктурах корпоративного рівня, створення на їх основі інформаційних технологій управління якістю ІТ-послуг в умовах консолідації інформаційно-обчислювальних ресурсів і використання засобів віртуалізації, та подальше використання цих ІТ у різноманітних системах управління ІТ-інфраструктурами для автоматичного управління рівнем якості ІТ-послуг, дозволить не лише підтримувати узгоджений рівень ІТ-послуг, а ще й раціонально використовувати інформаційно-обчислювальні ресурси, що задіяні для надання цих послуг. Тому дана робота, яка присвячена управлінню корпоративними ІТ-інфраструктурами задля забезпечення та підтримки заданого рівня якості ІТ-послуг, є актуальною.

Проблема, що вирішується в дисертації: розробка комплексу математичних моделей, методів, алгоритмів та інформаційних технологій для

аналізу та управління рівнем якості ІТ-послуг, що надаються корпоративними ІТ-інфраструктурами.

Зв'язок роботи з науковими програмами, планами, темами. В дисертаційну роботу включені основні результати, отримані автором в період з 2007 по 2021 рр. на кафедрі автоматики та управління в технічних системах Національного технічного університету України «Київський політехнічний інституту імені Ігоря Сікорського» в рамках наукового напрямку кафедри «Управління ІТ-інфраструктурою». Тематика роботи включена в науково-технічні плани Наукового парку «Київська політехніка», навчально-наукового центру «НТУУ«КПІ»-Неткрекер», кафедри автоматики і управління в технічних системах. Робота виконувалася в рамках таких держбюджетних і договірних науково-дослідних (НДР) і дослідно-конструкторських (ДКР) робіт: НДР №2143-ф «Розробка методів і засобів управління функціонуванням інформаційно-телекомунікаційних систем» (Держ. реєстр. роботи № НДР 0108U000490, Держ. обл. звіту № 0209U010830); НДР № 2302-ф «Розробка і дослідження математичних моделей та методів аналізу, синтезу і управління великими інформаційно-телекомунікаційними системами» (Держ. реєстр. роботи № НДР 0110U002195, Держ. обл. звіту № 0212U007761); НДР №2601-ф «Розроблення і дослідження моделей, методів та технологій проектування, програмування і управління хмарними ІТ-інфраструктурами» (Держ. реєстр. роботи № НДР 0113U002285, Держ. обл. звіту № 0216U005227); НДР №2844-п «Розробка та впровадження системи управління ІТ-інфраструктурою з консолідованими інформаційно-обчислювальними ресурсами» (Держ. реєстр. роботи № НДР 0115U000322); НДР №2919-ф «Платформа розроблення, експлуатації і розвитку критичних ІТ-інфраструктур для роботи з великими даними» (Держ. реєстр. роботи № НДР 0116U003801, Держ. обл. звіту № 0219U001317); НДР №2024 «Хмарна платформа розроблення і управління функціонуванням критичних ІТ-інфраструктур, що опрацьовують великі обсяги даних» (Держ. реєстр. роботи № НДР 0117U000537, Держ. обл. звіту № 0220U100804); Держконтракт (договір) на виконання ОКР № Т29-6.2/41 «ДКР - Калина-1к» від 01.02.2007; договорів № 1/562-09 від 29.01.2009 р, № 1/562-10

від 27.01.2010 р; № 1-577/11 від 19.07.2011 р та № 1/577-12 від 06.07.2012 р з ТОВ «НЕТКРЕКЕР». Всі роботи завершувалися створенням модулів інформаційної технології управління ІТ-інфраструктурою, макетів і дослідних зразків системи управління ІТ-інфраструктурою або її підсистем.

Мета і завдання дослідження. Метою роботи є забезпечення підтримки якості ІТ-послуг на заданому рівні шляхом розроблення та впровадження моделей, методів, інформаційних технологій та інструментів керування ІТ-інфраструктурою за умов раціонального використання її консолідованих інформаційно-обчислювальних ресурсів.

Для досягнення поставленої мети в роботі вирішуються такі завдання:

- проаналізувати проблеми управління якістю ІТ-послуг у корпоративній ІТ-інфраструктурі та встановити дії, які необхідно виконувати для підвищення конкурентоспроможності та ефективного виконання бізнес-процесів постачальників ІТ-послуг.

- проаналізувати властивості ІТ-послуг, які надають користувачам корпоративні ІТ-інфраструктури, та визначити показники функціонування компонентів ІТ-інфраструктури, які впливають на продуктивність застосунків, які надають відповідні послуги. Визначити за яких умов та яким чином є можливість керувати рівнем якості ІТ- послуг, що надаються ІТ-інфраструктурою.

- розробити метод визначення поточного рівня якості ІТ-сервісів на основі параметрів функціонування компонентів ІТ-інфраструктури, які задіяні для надання ІТ- послуг, що розглядаються. Визначити способи отримання або розрахунку значень параметрів обладнання, що впливають на якість ІТ- послуг.

- розробити алгоритм навчання нейронної мережі, що дозволить автоматично визначати структуру нейронної мережі.

- розробити метод визначення компонентів ІТ-інфраструктури, функціонування яких має відхилення від нормативного.

- розробити метод визначення поточного рівня якості ІТ-послуг що базуються на аналізі статистики, яка формується на основі параметрів функціонування елементів ІТ-інфраструктури.

– розробити метод управління якістю ІТ-послуг на основі даних моніторингу функціонування інформаційно-телекомунікаційних компонентів ІТ-інфраструктури, задля забезпечення узгодженого рівня якості сервісів, що надаються.

– розробити метод пошуку компонентів ІТ-інфраструктури, що надає ІТ-послуги, що мають відхилення від нормативного функціонування.

– розробити підхід, що дозволяє налаштовувати систему автоматичного управління ІТ-інфраструктурою. Даний підхід має дозволяти ІТ-підрозділу легко налаштовувати сценарії поведінки системи управління, які будуть задіяні при відхиленні рівня якості послуг, що надаються ІТ-інфраструктурою

– розробити метод, що дозволить автоматично керувати множиною гомогенних серверів та балансувати навантаження. Даний метод дозволить ІТ-інфраструктурі реагувати на збільшення користувачів сервісами, що надаються.

– на базі запропонованих методів розробити підхід до створення інформаційної технології управління ІТ-послугами, яка відрізняється від існуючих високою точністю оцінки рівня якості сервісів, що дозволить підвищити швидкість та продуктивність керуючих впливів при управлінні рівнем якості ІТ- послуг.

Об’єктами дослідження є ІТ-послуги, які надаються користувачам корпоративної ІТ-інфраструктури.

Предмет дослідження – методи та засоби управління рівнем якості ІТ-послуг, які забезпечують підтримку узгодженого рівня якості ІТ-послуг за умов раціонального використання ресурсів ІТ-інфраструктури корпоративного рівня.

Методи дослідження. Для визначення якості ІТ-сервісів використано апарат нечіткої логіки та нейромережеві класифікатори, а також апарат непараметричної статистики. Для визначення керуючих впливів, що забезпечать більш високу якість надаваних сервісів використано апарат нечіткої логіки. Було враховано методи управління цифровим обладнанням ІТ-інфраструктури, протоколи передачі даних по мережі та методи моніторингу обчислювального та комунікаційного обладнання ІТ-інфраструктури.

Наукова новизна полягає в наступному:

1. Вперше розроблено структурний метод навчання нейромережевих класифікаторів, який відрізняється від методу зворотного поширення похибки тим, що структуру нейронної мережі не треба визначати заздалегідь, оскільки вона динамічно підлаштовується під особливості задачі підтримки якості ІТ-сервісів на узгодженому рівні.

2. Вперше розроблено метод оцінки рівня якості ІТ-сервісів, який відрізняється використанням апарата непараметричної статистики для зведення в евклідовому просторі значень параметрів функціонування ІТ-інфраструктури, що впливають на рівень якості ІТ-сервісів.

3. Вперше запропоновано метод управління рівнем якості ІТ-послуг, який відрізняється тим, що за допомогою апарату нечіткої логіки та нейромережевих класифікаторів визначається взаємозв'язок між обсягами задіяних обчислювальних ресурсів та значеннями показників якості ІТ-сервісів, після чого підтримується узгоджений рівень якості сервісів регулюванням обсягів ресурсів ІТ-інфраструктури.

4. Отримала подальший розвиток інформаційна технологія управління рівнем якості ІТ-сервісів, яка відрізняється від існуючих тим, що дозволяє автоматично підтримувати якість сервісів на узгодженому рівні шляхом управління обсягами обчислювальних ресурсів ІТ-інфраструктури, які задіяні для надання ІТ-сервісів.

Практичне значення отриманих результатів полягає у тому, що:

- розроблена архітектура системи управління ІТ-інфраструктурою з урахуванням значущості бізнес-процесів;
- розроблені математичні моделі і алгоритми можуть бути використано в програмному забезпеченні управління ресурсами і навантаженнями ІТ-інфраструктури;
- теоретичні і практичні результати дисертаційної роботи використані при створенні автоматизованої системи управління ТОВ «СІТІУС ПРО» та ТОВ «АЙАДМІН», системи управління ІТС кафедри

автоматики та управління в технічних системах КПП ім. Ігоря Сикорського.

Отримані результати можуть бути використані для створення систем автоматичного управління ресурсами IT-інфраструктури. Також вивчено природу якості веб сервісів та визначено основні параметри, що впливають на якість таких послуг, розроблено методи автоматичного визначення значень таких параметрів.

Особистий внесок здобувача. У наукових працях, надрукованих у співавторстві, автору належать: у [1] – розробка методу оцінки якості функціонування елементів IT-інфраструктури з використанням зоноїдів та апарату нечіткої логіки; у [2] – запропоновано структурне навчання штучних нейронних мереж та використання побудованих штучних нейронних мереж при оцінці якості функціонування елементів IT-інфраструктури; [3] – аналіз можливості використання штучних нейронних мереж в якості класифікатора, який використовується для генерування керуючого впливу; [4] – аналіз можливості використання нейромережевих класифікаторів при визначенні якості сервісу VoIP, аналіз параметрів, що впливають на якість сервісу VoIP; [8] – розробка системи автоматичного управління елементами IT-інфраструктури, що базується на використанні агентів, що встановлюються на сервери та робочі станції; [5] – створення методу балансування навантаження, що базується на апараті нечіткої логіки (АНЛ) з використанням класифікатора на базі штучної нейронної мережі, розробка стенду HTTP сервісу для підтвердження працездатності запропонованого методу;

Апробації результатів дисертації. Основні результати дисертації оприлюднено у доповідях на конференції «Інформаційні технології – інструмент підвищення конкурентоздатності підприємств» (Київ 2009); міжнародній конференції “Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference” (Молдова, 2014); міжнародній конференції “CLOUD COMPUTING” (Афіни, 2017); міжнародній конференції “Problems of Infocommunications Science and Technology (PIC S&T), Second International Scientific-Practical Conference” (Харків, 2015); міжнародній конференції “Radio

Electronics & Info Communications (UkrMiCo)” (Київ, 2016); міжнародній конференції “Congress on Information Technology, Computational and Experimental Physics” (Польща 2015)

Публікації. За результатами досліджень опубліковано 18 наукових праць, у тому числі 6 статей у наукових фахових виданнях (з них 2 статті у періодичних наукових виданнях інших держав, які входять до ОЕСР та Європейського Союзу, 1 стаття в закордонних виданнях, що входять до Scopus та 3 статті у фахових виданнях), 12 тез доповідей і матеріалів конференцій (з них 3, що індексуються Scopus.

Структура та обсяг дисертації. Дисертаційна робота складається із вступу, чотирьох розділів, висновків та списку використаних джерел (99 найменувань) на 130 сторінках. Загальний обсяг роботи – 154 сторінки, основний текст роботи викладено на 120 сторінках. Робота містить 43 рисунок та 6 таблиць.

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМ УПРАВЛІННЯ ЯКІСТЮ ІТ-ПОСЛУГ

1.1 Особливості ІТ-послуги як об'єкту управління

Інформаційні технології (ІТ) – це технології, що використовують комп'ютери і відповідне програмне забезпечення (ПЗ) для обробки інформації. Інформаційні технології та системи передбачають збір, передачу, перетворення, зберігання, захист, відображення та використання інформації. Інформаційно-комунікаційні технології (ІКТ) – це технології, що використовують мережеве обладнання, комп'ютери та необхідне програмне забезпечення для передачі інформації.

Інформаційні та інформаційно-комунікаційні технології – невід'ємна частина сучасного бізнесу, а підвищення ефективності всіх бізнес-процесів досягається за рахунок їх автоматизації виконання бізнес-операцій, що забезпечується ІТ-послугами.

Для підтримки бізнес-процесів або процесів діяльності підрозділів підприємства, йому надається сукупності різноманітних ІТ-послуг, що об'єднуються у ІТ-сервіси:

$$\begin{aligned} S &= \{S_o \mid o = \overline{1, \dots, O}\}, \\ E &= \{E_{oe} \mid e = \overline{1, \dots, E_o}, o = \overline{1, \dots, O}\}, \end{aligned} \quad (1.1)$$

де S – множна ІТ-сервісів, що надаються користувачам, O – кількість сервісів, що надаються користувачам, E – множна всіх ІТ-послуг, що надаються користувачам, E_o – кількість послуг, що входить до складу ІТ-сервісу S_o , E_{oe} – e -та послуга, що входить до складу o -го сервісу, $o \in [1, O]$, $e \in [1, E_o]$.

Створення, розгортання, впровадження та експлуатація ІТ-послуг вимагає вирішення низки питань, насамперед, визначення: конкретних процесів, підлягаючих автоматизації в першу чергу, відповідальності замовника та постачальника ІТ-послуг, вимог до процесів забезпечення функціонування ІТ-

послуги, переліку обладнання та сукупності застосунків, необхідних для надання послуги, витрат (тарифних планів) на послуги, правил використання ІТ-послуг. Набір ІТ-сервісів, що є необхідними підприємству є індивідуальним і залежить від галузі, розмірів підприємства, рівня його автоматизації, кваліфікації персоналу, стратегії розвитку, тощо.

Прибуток та витрати підприємства безпосередньо залежать від рівня автоматизації виконання бізнес-операцій та якості роботи наданих йому ІТ-сервісів. Тому необхідно постійно контролювати функціонування ІТ-інфраструктури та якість надання ІТ-послуг. Висока якість надаваних ІТ-послуг може буди забезпечена за рахунок великої кількості обчислювальних ресурсів ІТ-інфраструктури, які виділені застосункам для надання ІТ-послуг. Але такий підхід потребує великих фінансових витрат, оскільки приводить до нерационального використання ресурсів ІТ-інфраструктури.

Задача забезпечення якості надаваних ІТ-послуг лежить на їх постачальниках – спеціалізованих ІТ-підрозділах, всі функції яких можна розділити на три великі групи:

- забезпечення функціонування ІТ-інфраструктури;
- підтримка бізнес-застосувань;
- підтримка та обслуговування користувачів.

Активи ІТ-підрозділів, якими потрібно управляти для створення та управління якістю ІТ-послуг включають в себе персонал, процеси, обладнання/технології, користувачів/постачальників.

Між бізнес-підрозділами та ІТ-підрозділами укладається договори або домовленості про надання ІТ-сервісів та рівень якості таких сервісів. Угода про рівень обслуговування (SLA) – це зобов'язання між постачальником послуг і користувачем. Найважливіші аспекти надання ІТ-послуг – якість послуг, її наявність та відповідальність вимогам – узгоджуються між постачальником послуг та користувачем послуги [6]. Як правило, інтернет-провайдери та телекомунікаційні компанії, формулюють угоди про рівень послуг в рамках своїх контрактів з користувачами та визначають рівень якості послуги. У цьому

випадку, як правило, SLA містить значення середнього часу між відмовами (MTBF), середнього часу для відновлення або середнього часу до відновлення (MTTR); визначення того, яка сторона несе відповідальність за несправності або сплату комісій. Постачальникам ІТ-сервісів не вигідно надавати продукт, якість якого перевищує узгоджений рівень адже в такому випадку постачальники послуг несуть фінансові збитки через використання надмірної кількості ресурсів.

Провідні компанії світу створюють системи управління

Таблиця 1.1 Порівняння існуючих систем управління ІТ-інфраструктурою

	CISCO	NetCracker	Kubernetes	Amazon
Управління обладнанням	Присутнє	Часткова інтеграція під замовлення	Відсутнє	Використання свого обладнання
Моніторинг обладнання	Присутній	Присутній	Лише обладнання, що використовується	Лише обладнання, що використовується
Моніторинг якості сервісів	Опосередковано через моніторинг самого обладнання	Через моніторинг обладнання	Через моніторинг ресурсів, що задіяні для забезпечення послуги	Через моніторинг ресурсів, що задіяні для забезпечення послуги
Автоматичне управління якістю сервісів	Опосередковано через управління обладнанням	Через систему правил поведінки	Через управління віртуальними машинами	Через управління віртуальними машинами

Як видно з таблиці 1 провідні компанії в галузі управління ІТ-інфраструктурами розробляють системи управління ІТ-інфраструктурою, основані на моніторингу та аналізі функціонування обчислювального та комунікаційного обладнання. Тобто використовується правило: якщо обладнання функціонує на належному рівні, то це обладнання надає ІТ-послуги необхідного рівня.

В даній роботі розглядається підхід до управління ІТ-інфраструктурою, що базується на управлінні якістю ІТ-послуг, що надаються даною ІТ-інфраструктурою, а не на управлінні, спрямованому на підтримку її функціонування. Такий сервісно-орієнтований підхід до управління ІТ-інфраструктурою дозволяє більш ефективно керувати рівнем якості ІТ-послуг, адже на якість послуг впливає не лише якість функціонування обладнання, а й інші фактори.

На рисунку 1.1 схематично виділено три рівні, на яких можна аналізувати якість сервісів, що надаються ІТ-інфраструктурою.

Основною метою діяльності постачальників ІТ-сервісів є забезпечення максимальної задоволеності користувачів ІТ-послуг при мінімальних затратах ІТ-ресурсів. Для досягнення цієї мети необхідно вирішувати задачі перерозподілу ресурсів між ІТ-послугами та визначати доцільність надання менш важливих послуг з високою якістю.

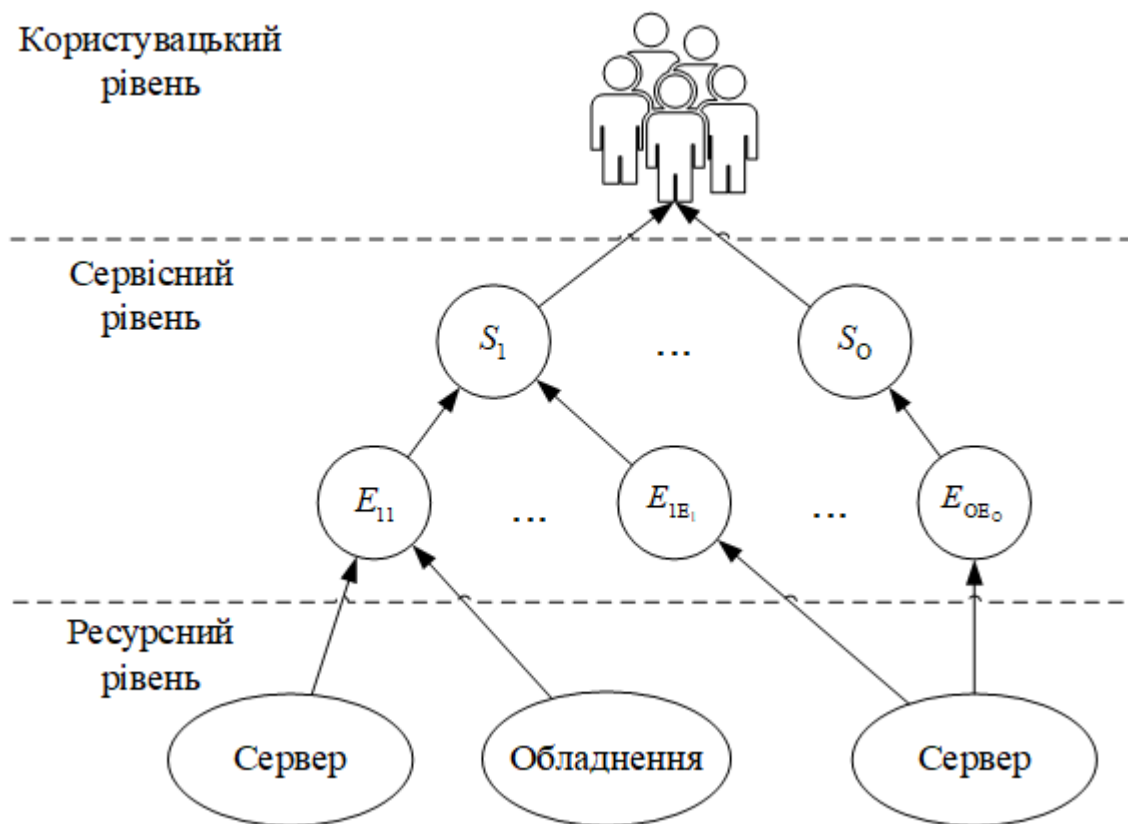


Рисунок 1.1 Рівні аналізу сервісів

На рівні користувачів є можливість аналізувати сервіси на основі користувацьких відгуків, звернень до ІТ-підрозділів, заявок на розширення чи зміну існуючих сервісів. Ця інформація дозволяє ІТ-підрозділу виконувати стратегічне управління якістю сервісів завдяки аналізу проблем, які виникають в ІТ-інфраструктурі. Здійснювати модернізацію та планування розвитку ІТ-інфраструктури.

На сервісному рівні ІТ-підрозділ аналізує показники функціонування складових кожного сервісу – ІТ-послуг, які входять до складу кожного сервісу. Оскільки вимоги до якості сервісів чи умови надання сервісів з часом можуть змінюватися, то ІТ-підрозділам необхідно мати засоби автоматизованого налаштування сервісів та відповідних ІТ-послуг.

На рівні ресурсів ІТ-підрозділи аналізують функціонування програмних та апаратних ресурсів, що забезпечують роботу сервісів. За необхідності можуть приймати рішення про переналаштування, заміну чи перерозподіл ІТ-ресурсів.

Базуючись на результатах цих трьох рівнів аналізу надання сервісів, ІТ-підрозділи визначають дії, які необхідно виконати на стратегічному та оперативному рівні управління ІТ-інфраструктурою для забезпечення якості надаваних ІТ-сервісів на узгодженому рівні. Інформація, яка отримується на користувацькому та сервісному рівнях також використовується для аналізу якості надання сервісів менеджерами ІТ-підрозділу та автоматизованими системами CRM.

В роботі вирішується задача оперативного управління якістю ІТ-сервісів при незмінних вимогах до сервісів, що надаються ІТ-інфраструктурою. Для цього необхідно здійснювати безперервний моніторинг компонентів ІТ-інфраструктури, які забезпечують функціонування надаваних ІТ-сервісів. Отримана інформація моніторингу дає можливість визначати керуючі впливи для підтримання якості сервісів на узгодженому рівні. Також, за умови, що якість надаваних сервісів не падає нижче оговореного рівня, постачальники ІТ-сервісів мають змогу вивільняти надлишкові обчислювальні ресурси та використовувати вивільнені ресурси для роботи інших ІТ-послуг, або відключати їх для економії електроенергії. Для вирішення цієї задачі необхідно розробити відповідну

інформаційну технологію управління IT-інфраструктурою. Системи управління IT підприємств є складними, оскільки потрібно здійснювати моніторинг, облік, аналіз показників функціонування усіх бізнес-процесів.

Усі IT-сервіси можна розбити на дві категорії [7]: бізнес-сервіси та технологічні сервіси. Бізнес-сервіси використовуються для автоматизації виконання бізнес-процесів та бізнес-операцій і залежать від специфіки бізнесу підприємства. Ці сервіси надають ERP, PLM, CRM та інші системи, що дозволяють керувати безпосередньо самими бізнес-процесами. Технологічні сервіси надаються технологічними підсистемами, що забезпечують роботу IT-систем. До них відносяться сервіси інформаційної безпеки, електронна пошта, відео-зв'язок, веб-сервіси, сервіси передачі файлів, IP-телефонія, засоби забезпечення та контролю доступу до інтернету, служби миттєвого обміну повідомленнями, тощо.

Логіка роботи більшості бізнес-сервісів доволі складна, залежить від належного функціонування багатьох компонентів IT-інфраструктури, зазвичай прихована від користувачів і впливати на якість роботи безпосередньо цих сервісів вкрай важко. Але завжди існує залежність між якістю надання бізнес-сервісів та обсягом ресурсів, що виділено для їх надання, кількістю користувачів, значень показників функціонування IT-інфраструктури тощо. Тому підтримку якості надання цих сервісів на узгодженому рівні можна забезпечувати шляхом забезпечення перерозподілу ресурсів IT-інфраструктури та балансування навантаження між зайнятими наданням бізнес-сервісами серверами.

Якість технологічних сервісів залежить від обладнання та IT-ресурсів, що забезпечують їх роботу. Специфіка функціонування технологічних сервісів дозволяє системам автоматичного управління IT-інфраструктурою управляти якістю таких сервісів шляхом управління обсягами ресурсів IT-інфраструктури, які надаються цим сервісам.

1.2 Аналіз проблем визначення якості ІТ-сервісів

Згідно поточної версії ДСТУ ISO 9000 [26], «якість» – це ступінь відповідності сукупності властивих характеристик вимогам. В свою чергу, стандарт ISO 9001 [27] установлює вимоги до системи управління якістю, що їх можна застосовувати для внутрішніх цілей організації або для сертифікування чи укладання контрактів. У ньому зосереджено увагу головним чином на результативності системи управління якістю для задоволення вимог замовника послуг.

Задачею систем автоматичного управління рівнем якості ІТ-сервісів є забезпечення відповідності фактичних значень показників надання усіх сервісів значенням цих показників, які зафіксовано у SLA. Але основним узагальненим критерієм якості роботи ІТ-сервісів можна вважати рівень задоволеності користувача. Для визначення рівня задоволеності користувачів ІТ-сервісом необхідно визначити залежність рівня задоволеності сервісом від якості ІТ-послуг, що є складовими цього сервісу. Для визначення якості ІТ-послуг необхідно задати критерії, за якими оцінюється якість послуги та характеристики, за допомогою яких можна оцінювати такі критерії.

Для прикладу, на даний момент існує стандарт ISO/IEC 25010 [22] що налічує вісім критеріїв, за якими є можливість оцінити якість програмного забезпечення:

- функціональна придатність (functional suitability);
- зручність використання (usability);
- надійність (reliability);
- зручність супроводу (maintainability);
- переносимість (portability);
- ефективність роботи (performance efficiency);
- сумісність (compatibility);
- безпека (security).

Для порівняння стандарт ISO 9126:2001 [23] 2001 року описує лише шість критеріїв

- функціональність (functionality);
- зручність використання (usability) або практичність;
- надійність (reliability);
- зручність супроводу (maintainability);
- переносимість (portability);
- продуктивність (efficiency) або ефективність.

Як видно при порівнянні, частина критеріїв трохи змінилася, та було додано ще два критерії: сумісність та безпека. Такі зміни полягають у тому, що вимоги до програмного забезпечення та його експлуатація з часом змінюються. Такі зміни є невід’ємною частиною розвитку ІТ індустрії, здешевшання виробництва та зростання потужностей апаратного забезпечення.

Так як якість сервісів тісно пов’язана із якістю програмного забезпечення що надає ці сервіси, а різноманітність надаваних ІТ-сервісів величезна, то у даній роботі для оцінки якості ІТ-сервісів взяті за основу саме критерії, що перелічені у стандарті ISO/IEC 25010 [22].

Слід зазначити, що при автоматичному керуванні рівнем якості сервісів немає можливості підвищувати такі критерії як зручність супроводу, зручність використання, переносимість, сумісність. Також у роботі лише частково розглядаються такі критерії як надійність, безпека та функціональна придатність. Це зумовлено тим, що системи автоматичного керування рівнем якості ІТ-сервісів не можуть покращувати роботу програмного забезпечення, а можуть лише давати можливість якісно виконувати те, що було написано та спроектовано розробником. Однією із задач системи, що розробляється, є унеможливлення зменшення або аргументоване часткове зменшення рівня якості ІТ-сервісів за окремими критеріями.

Параметри якості – це кількісні або якісні характеристики сервісу. Кількісні параметри – це параметри, що піддаються вимірюванню. В свою чергу, показники, або метрики – це значення параметрів, що розраховане відповідно до

замірів. Якісні параметри – це параметри, що можна задати лише за допомогою людського судження.

Відповідно до ДСТУ ІТУ-Т Е.800:2014 [24] існує чотири точки зору щодо якості надаваної послуги. Схематично вони зображені на рисунку 1.2.

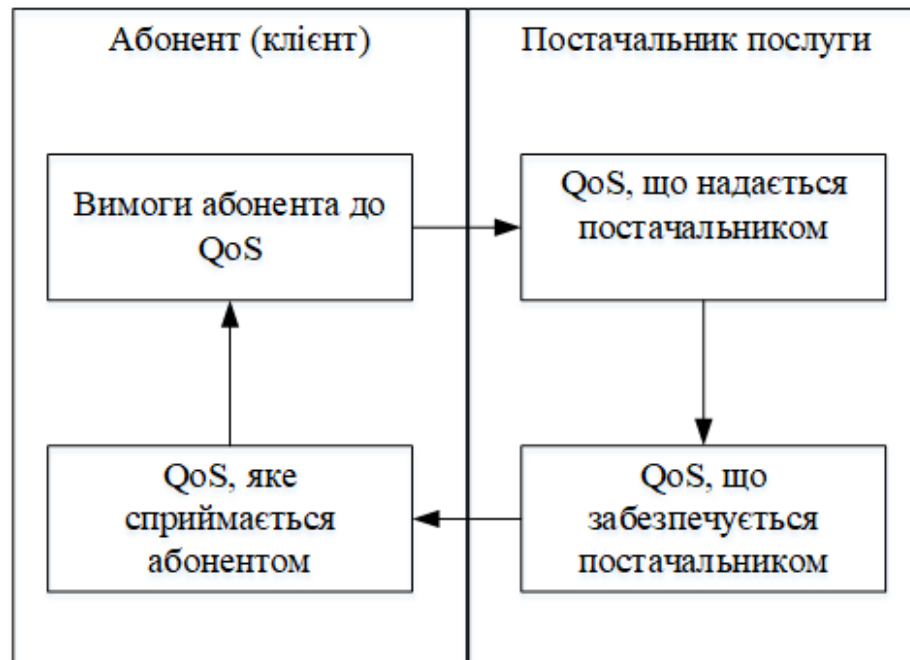


Рисунок 1.2 Сприйняття якості обслуговування

Даний рисунок ілюструє взаємодію користувача та постачальника. Задачею для розробника систем автоматичного керування рівнем якості ІТ-сервісів є розробка системи що дасть змогу керувати елементами ІТ-інфраструктури, що забезпечує функціонування ІТ-сервісів, таким чином щоб робота ІТ-сервісів відповідала SLA. Це дає змогу постачальнику відповідати рівню якості. Але постачальник послуг може використовувати більше ресурсів ІТ-інфраструктури ніж це потрібно для забезпечення відповідного рівня якості послуг. Це може призводити до надмірних фінансових витрат для постачальника послуг. Тому не менш важливою задачею таких систем управління є зменшення вартості надання послуг для самого постачальника. Це дасть змогу постачальнику мінімізувати витрати на надання послуг при умові що рівень QoS буде сприйматися користувачем як належний. В роботі не розглядається можливість що навіть повна відповідність SLA може не задовольняти окремого

користувача, адже сприйняття якості роботи ІТ-сервісу є індивідуальним для кожної людини. Таку проблему не можливо вирішити за допомогою автоматичних систем керування, адже основою для них є саме відповідність SLA.

Задачу визначення рівня задоволеності користувача якістю сервісів, що надаються можна звести до задачі визначення залежності якості надаваних сервісів від значень окремих параметрів якості (рис. 1.2).

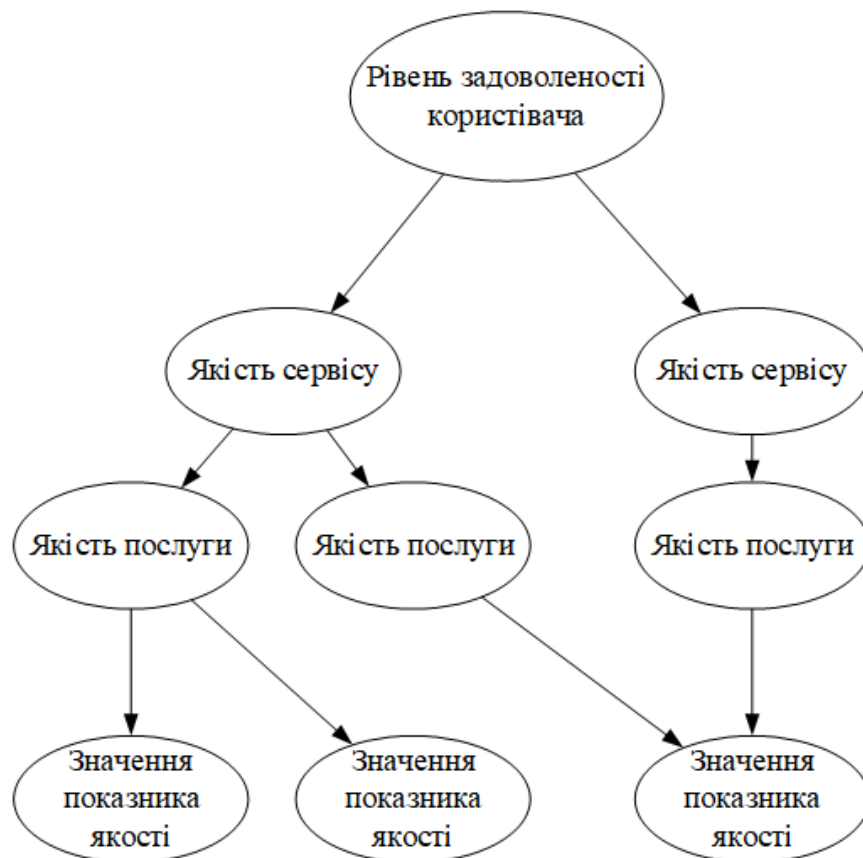


Рисунок 1.3 Залежність рівня задоволеності користувача від якості надаваних послуг

На рисунку 1.3 схематично зображено залежність рівня задоволеності користувачів сервісом від якості роботи складових сервісу.

При управлінні якістю надаваних сервісів доцільно виділити дві задачі, які треба вирішити: управління застосунками, що забезпечують роботу ІТ-сервісу та управління ресурсами, що використовуються для надання сервісу. Задача управління застосунками зводиться до їх конфігурації відповідно до бізнес-

задач, що вони вирішують. Системи автоматичного управління ІТ-інфраструктурою мають виконувати переконфігурацію таких застосунків при зміні бізнес-задач. На якість роботи застосунків також впливають і ІТ-ресурси, що для них виділено. Тому при управлінні ресурсами ІТ-інфраструктури системи управління опосередковано управляють якістю самих застосунків (рисунк 1.4).

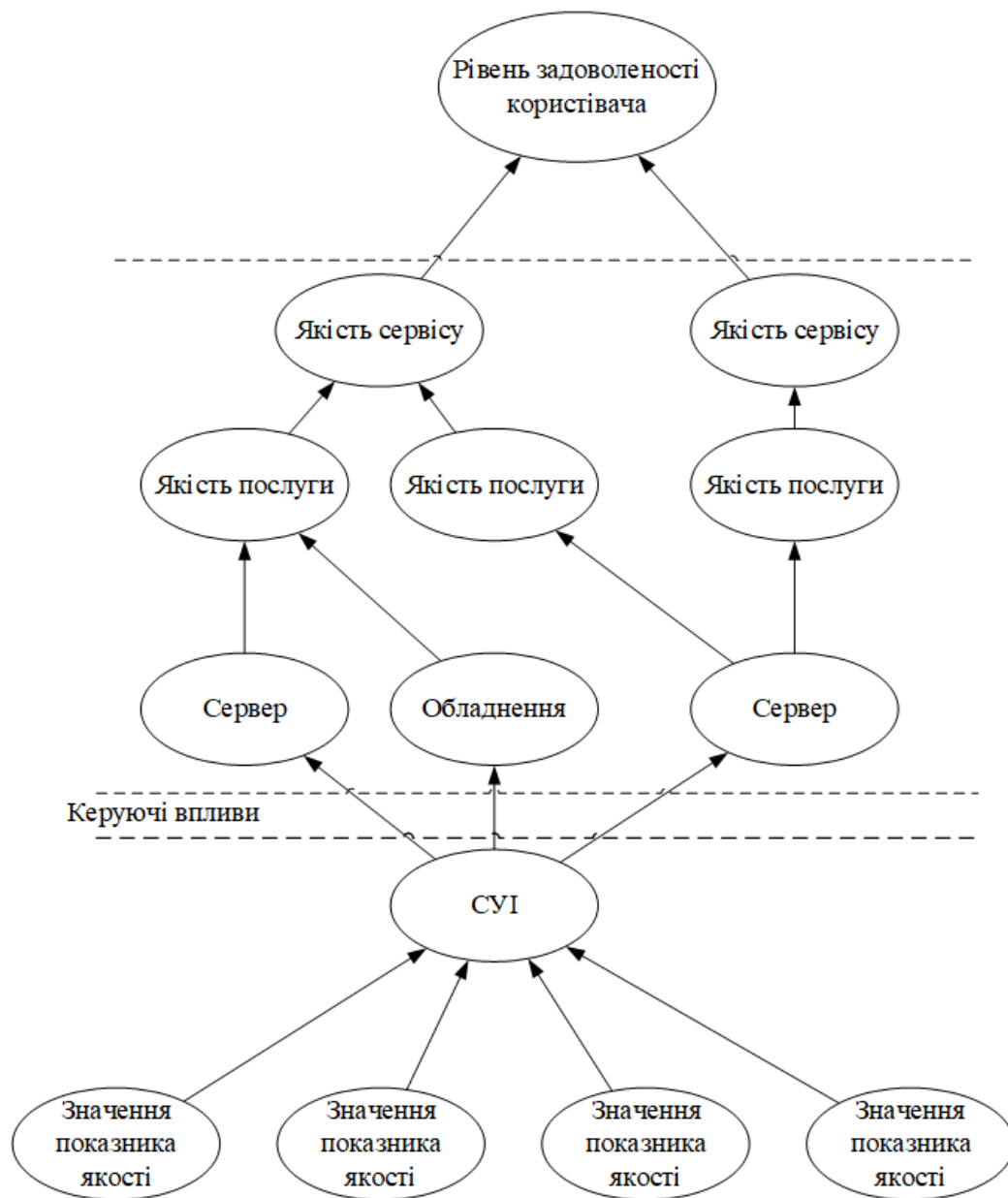


Рисунок 1.4 Вплив керуючих дій на рівень якості надаваних сервісів

Базуючись на значення показників якості надаваних ІТ-послуг системи автоматичного управління мають змогу визначати керуючі впливи, що треба задіяти для забезпечення якості надаваних сервісів на узгодженому рівні.

Керуючі впливи, що має виробляти така система зазвичай закладені в OMS-системах та задаються саме при розробці таких OMS-систем. Специфіка OMS-систем сильно залежить від бізнес-процесів та є унікальною для кожної окремо взятої фірми. Вирішення задачі безпосереднього управління застосунками не розглядається в даній роботі, адже вона відноситься до задач розробників самих застосунків, а не до задач автоматичного управління рівнем якості надаваного ІТ-сервісу.

Для вирішення задачі оцінки якості надаваного сервісу, необхідно вирішити наступні задачі: визначити параметри ІТ-інфраструктури, що впливають на якість послуги, визначити способи вимірювання значень таких параметрів, визначити спосіб зведення значень параметрів до узагальненої оцінки якості роботи ІТ-сервісу.

Зведена оцінка якості залежить не тільки від кількісних, а й від якісних параметрів. Тому доцільно при розрахунках використовувати апарат нечіткої логіки, який дозволяє оперувати зрозумілими людині значеннями.

Особливий інтерес представляє QoS, що оцінюється користувачем (так зване QoSE або QoSP). На QoSE впливає якість сервісу, що надається, а на сприйняття користувача – психологічні чинники. Розуміння QoSE має першорядне значення для сприяння оптимізації доходів і ресурсів постачальника послуг.

Вимоги до QoS користувача (QoSR) – це вимоги, до єдиних вимог до якості або потребі у ньому, що можуть бути виражені описовими термінами (критеріями), перерахованими в порядку пріоритету з вказаним значенням якості для кожного критерію. Постачальник послуг переводить їх потім в параметри і метрики, що стосуються обслуговування.

QoS, пропоноване постачальником послуг (QoS_O) – це заява про рівень якості, планованого і тому пропонованого користувачу постачальником послуг. Рівень QoS, який постачальник послуг передбачає забезпечити, а, отже, пропонує користувачеві, виражається цільовими значеннями або діапазоном для оцінки параметрів, що стосуються конкретної послуги.

QoS, що надається або досягається постачальником послуг (QoSD) – це заява про рівень QoS, що надається користувачу. QoS, що забезпечується або надається виражається метриками для відповідних параметрів тієї чи іншої послуги.

QoS, що сприймається користувачем (QoSE) – це заява про рівень якості, яке, на думку користувачів, вони відчують. Рівень QoS, що відчувається користувачем може бути виражений за допомогою оцінки думок. QoSE складається з двох основних складових – кількісної та якісної. Кількісна складова може залежати від впливів на всю наскрізну систему (інфраструктуру мережі), а якісна складова може залежати від очікувань користувачів, умов довкілля, психологічних факторів, прикладного контексту і ін. QoSE може також розглядатися як QoSD, що тлумачиться користувачем при відповідних якісних факторах, що впливають на сприйняття послуги.

Для побудови автоматизованої системи управління якістю ІТ-сервісу, критерії якості доцільно розділити на три групи [8]: критерії обслуговування, вартісні критерії та технічні.

Критерії обслуговування визначаються окремо для кожного ІТ-сервісу. Це можуть бути такі якісні критерії як зручність використання або технічна підтримка та ін. Вартісні критерії також є індивідуальними для кожного ІТ-сервісу – це абонентська плата, ціна підключення, ціна проектування, тощо.

Технічні критерії якості ІТ-сервісів можуть бути: доступність, ефективність, тощо. Для прикладу, доцільно розглядати такі технічні критерії якості сервісу VoIP [8]: якість голосу, затримка з'єднання, затримка при передачі та надійність зв'язку. Ці критерії можна задати або лінгвістичними значеннями, або числовими. Їх значення визначається базуючись на технічні ІТ-характеристики сервісу, ресурси, що задіяні для роботи сервісу та від програмного забезпечення, що використовується для надання сервісу.

Якість ІТ-послуг включає показники роботи мережі і показники, які не відносяться до роботи мережі. Прикладами параметрів роботи мережі є коефіцієнт помилок по бітам, запізнювання і ін., а прикладами показників, що не відносяться до роботи мережі, - час надання, тривалість ремонту, діапазон

тарифів та час вирішення скарг. Список критеріїв якості IT-інфраструктури для конкретної послуги буде залежати від послуги, і їх значення може змінюватися в залежності від сегментів сукупності користувачів.

Під своєчасністю мається на увазі наявність запасів ресурсів в потрібному для споживача час, можливість персоналу допомогти користувачу, швидкість здійснення послуг, доречність часу виконання послуги для користувача.

Обсяг переданих даних характеризує якість сервісу зі сторони надмірності використання ресурсів як замовника послуги, так і її виконавця. Наприклад, при наявності великої кількості помилок при передачі даних від користувача, відбувається повторне надсилання даних, що спричиняє надмірне використання трафіка.

Цілісність даних – повнота інформації, що надходить від або до замовника послуги.

До параметрів, що впливають на якість деяких IT-сервісів можна віднести [9]:

- параметри часу: час обробки сервером запиту користувача, час очікування відповіді користувачем, затримка;
- параметри, що характеризують втрати даних: коефіцієнт втрат даних, значення джиттеру;
- параметри телекомунікаційного обладнання: швидкість передачі даних, смуга пропускання каналу зв'язку;
- параметри на стороні користувача: величина переданих/отриманих даних, актуальність даних, тощо.

1.3 IT-інфраструктури постачальника IT-сервісів

Під поняттям IT-інфраструктури розуміється телекомунікаційна розподілена обчислювальна інфраструктура, що поєднує в собі сукупність серверів, систем зберігання даних, програмних застосунків, ПК та телекомунікаційної мережі [7].

Технічне забезпечення ІТ-інфраструктури складається з серверів, персональних комп'ютерів, систем зберігання даних, мережі і комунікаційних програм. Програмне забезпечення характеризується операційними системами, інструментальними засобами розробки завдань, програмами підтримки систем управління і засобами забезпечення інформаційної безпеки, додатками – спеціальними програмами, які забезпечують підтримку бізнес-процесів підприємства і працездатність окремих автоматизованих робочих місць.

При розгляді питання управління елементами ІТ-інфраструктури в контексті забезпечення узгодженого рівня якості сервісів, що надає інфраструктура, доцільно розділяти ІТ-інфраструктуру на декілька ієрархічних рівнів, в залежності від специфіки сервісу, що розглядається. Так, в [10] при розгляді метрик операторів телекомунікаційних сервісів (ОТС) виділяється п'ять ієрархічних рівнів: телекомунікаційної технології, мережі, сервісів, абонентський і телекомунікаційного бізнесу. Слідуючи моделі OSI [11], концепція системи управління мережами (TMN) розглядає мережеву логічну архітектуру, що включає в себе п'ять рівнів управління [12] – це рівень мережевих елементів, рівні управління: елементами, мережею, послугами та рівень бізнес-управління. В [13] концепція управління ІТ-інфраструктурою розглядається зі сторони ієрархічної структури, що містить три шари: нижній – управління мережами, середній – управління системами, верхній – управління сервісом ІТ.

В ІТ-інфраструктурі – сукупності ІТ-системи і телекомунікаційної мережі [14], пропонується виділити чотири ієрархічних рівня: бізнес-застосунків, універсальних сервісів, обчислювальних ресурсів і мережевої взаємодії. У свою чергу, при необхідності, кожен з чотирьох рівнів може бути розділений на підрівні.

На рівні бізнес-застосунків працюють додатки, які мають безпосереднє відношення до виконання бізнес-процесів або процесів діяльності. Самі ці застосунки надають ІТ-послуги користувачам. На цьому рівні працюють системи: планування ресурсів підприємства (ERP), управління персоналом (HRM), управління взаємовідносинами з користувачами (CRM), управління

інформацією про товари (PLM), управління робочими процесами (Work Flow), документообігом, BSM і безліч інших ІТ, необхідних для успішного ведення бізнесу підприємства.

До рівня універсальних сервісів віднесені сервіси, які не залежать від специфіки бізнесу підприємства. Такими сервісами і ПО є: електронна пошта, СУБД, відеоконференц-зв'язок, різні веб-сервіси, сервіс передачі файлів, що включає обмін файлами за допомогою FTP і пірінгових мереж, IP-телефонія, служба миттєвого обміну повідомленнями (Instant messenger, IM), засоби забезпечення і контролю доступу в інтернет тощо.

Рівень обчислювальних ресурсів включає в себе ПК користувачів та ін., а також ресурси ЦОД. Для малих підприємств, яким немає потреби обслуговувати ЦОД, а даний рівень входять сервери, на яких розміщують серверні частини ПО рівнів бізнес-додатків і універсальних сервісів. Все або тільки частина обчислювальних ресурсів можуть бути реалізовані у вигляді хмарних рішень. При використанні зовнішніх хмар ІТ-підрозділ вирішує організаційні питання і здійснює контроль якості надання сервісів.

За допомогою рівня мережевої взаємодії забезпечується доступ користувачів до елементів рівня обчислювальних ресурсів.

Управління ІТ-інфраструктурою здійснюється за допомогою СУІ, призначеної для управління доступом і розподілом обчислювальних і телекомунікаційних ресурсів, контролю, аналізу поведінки і підтримки користувачів, автоматизації діяльності, тощо. СУІ дозволяє не тільки зменшити витрати на утримання ІТ-підрозділу, а й є засобом автоматизації роботи адміністраторів і керівників ІТ-підрозділу. Таким чином, СУІ не тільки автоматизує роботу адміністраторів з управління ІТ-інфраструктурою, але і вирішує завдання автоматизації управління ІТ-підрозділом.

Обов'язковою складовою ІТ-інфраструктури підприємств середнього і великого бізнесу є ІТ-підрозділ. Подробиці надання ІТ-послуг регламентуються пакетом угод про рівень сервісу (SLA), укладених між бізнес-користувачами та ІТ-підрозділом. Для підтримки якості ІТ-сервісів на зафіксованому в SLA рівні, адміністратори забезпечують безперервне функціонування ІТ-інфраструктури,

здійснюють її обслуговування і ремонт. При цьому адміністратори можуть управляти елементами ІТ-інфраструктури як за допомогою СУІ, так і безпосередньо вручну. Стандарт ISO 20000 [15, 16] підкреслює важливу роль СУІ, що підтримує управління ІТ-послугами, для контролю метрик, що характеризують поведінку ІТ-процесів.

Відповідно до ITIL в управлінні ІТ-послугами беруть участь мінімум два відділення ІТ-підрозділу: підтримки сервісів (Service Desk), що забезпечує підтримку користувачів, і відділення управління ІТІ (ICT Infrastructure Management Team), що відповідає за функціонування ІТС.

1.4 Особливості віртуалізації при управлінні ресурсами ІТ-інфраструктури

Віртуалізація – створення віртуального, тобто штучного, об'єкта чи середовища [17, 19, 28, 18]. З точки зору бізнесу є безліч причин для використання віртуалізації. Більшість переваг, що одержуються впливають з того, що прийнято називати консолідацією серверів. Якщо є можливість перевести деякі системи, що не повністю використовуються, на один сервер, то буде отримана істотна економія простору, споживання енергії, охолодження і спрощує адміністрування в силу того, що потрібно підтримувати меншу кількість серверів. Оскільки важко визначити ступінь використання сервера, технологія віртуалізації підтримує можливість живої міграції. Жива міграція дозволяє операційній системі і її додаткам бути переміщеними на новий сервер для балансування навантаження на наявне обладнання.

Віртуалізація важлива також для розробників. Ядро Linux працює в загальному адресному просторі, що означає, що збій в роботі ядра або будь-якого драйвера призводить до краху всієї операційної системи. Віртуалізація означає, що запускаючи множину операційних систем, і, якщо одна з них руйнується через якийсь помилки, гіпервізор і інші операційні системи продовжують працювати. Це може зробити налагодження ядра подібної налагодженні призначених для користувача додатків [18].

Термін «віртуалізація» часто використовується в комп'ютерних технологіях для позначення абстракції комп'ютерних ресурсів. Відповідно, він може стосуватися різних випадків.

У випадку обладнання – це емуляція. У цьому методі в хост-системі створюється віртуальна машина, яка моделює якусь іншу апаратну архітектуру.

У разі віртуалізації операційної системи існує три типи віртуалізації: програмна та апаратна та віртуалізація на рівні операційної системи.

Програмних віртуалізацій буває три типи:

- паравіртуалізація – спосіб віртуалізації, який представляє програмний інтерфейс подібний до, але не ідентичний апаратному забезпеченню, що вимагає адаптації гостьової операційної системи;
- вбудована віртуалізація;
- динамічна рекомпіляція – при використанні динамічної компіляції програма-емулятор перетворює фрагменти програми, що виконується в код, який може бути виконаний на іншому комп'ютері, безпосередньо під час її роботи.

Апаратна віртуалізація – повна віртуалізація з підтримкою спеціальної процесорної архітектури. На відміну від програмної віртуалізації, за допомогою даної техніки можливе використання ізольованих гостьових систем, керованих гіпервізором безпосередньо. Гостьова система не залежить від архітектури хостової платформи і реалізації платформи віртуалізації.

Віртуалізація на рівні операційної системи – метод, який дозволяє операційній системі створювати декілька користувачьких.

Одним з самих важких методів віртуалізації є емуляція апаратури. У цьому методі в хост-системі створюється віртуальна машина, яка моделює іншу апаратну архітектуру, як це показано на рисунку 1.1.

Недоліком цього виду віртуалізації є те, що всі процеси на гіпервізорі відбуваються повільно. Кожна команда повинна бути змодельована стосовно до реального апаратного забезпечення, а при цьому 100-кратне уповільнення являється звичайною справою. При повному моделюванні, яке має на увазі точне

виконання циклів, моделювання конвеєрів ЦПУ і поведінки кешу, реальна швидкість виконання може бути і в 1000 разів менше, ніж на реальному моделюється апаратурі. Плюси даної віртуалізації в тому, що користувач стає незалежним від виробника свого апаратного забезпечення. Для прикладу, можна земулювати роботу процесу Intel на процесорі AMD.

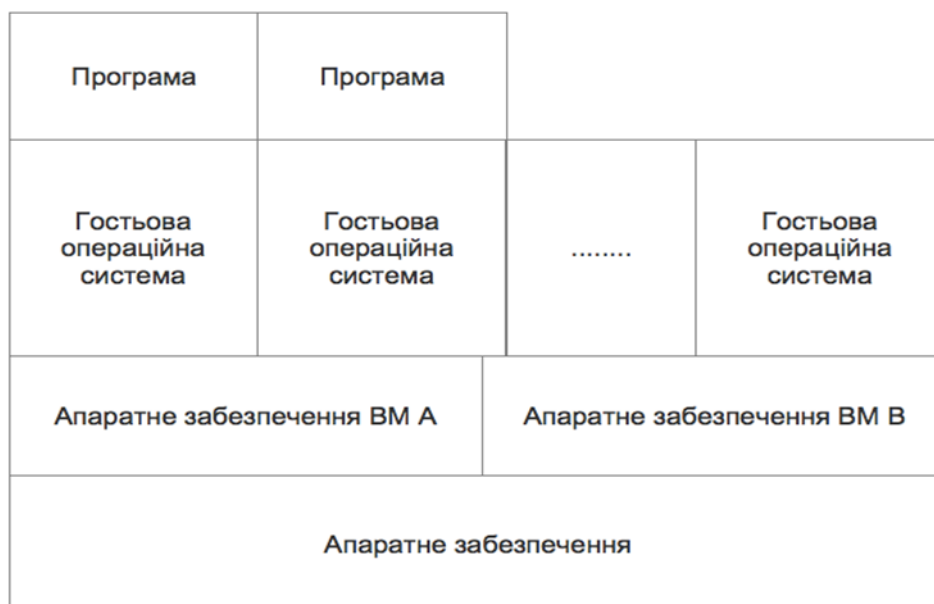


Рисунок 1.5 Схема емуляції апаратури

Повна віртуалізація використовує віртуальну машину, яка виступає як посередник між гостьовою операційною системою і реальним обладнанням. Схему даного типу віртуалізації наведено на рисунку 1.2. Слово "посередник" в даному випадку грає ключову роль, оскільки гіпервізор розташовується між гостьовою операційною системою і реальним обладнанням. Деякі інструкції захищеного режиму повинні перехоплюватися і оброблятися всередині гіпервізору, оскільки апаратура недоступна безпосередньо з операційної системи, доступ до неї надається через гіпервізор [19].

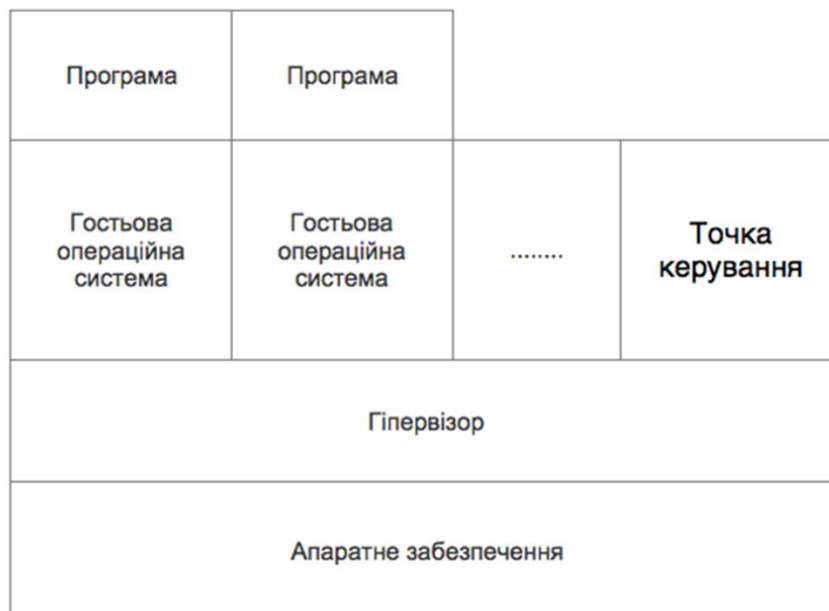


Рисунок 1.6 Схема повної віртуалізації

Повна віртуалізація працює швидше, ніж емуляція апаратури, але продуктивність все ж нижче, ніж на реальному обладнанні, через посередництво гіпервізору. Найбільша перевага повної віртуалізації полягає в тому, що гостьова операційна система може бути запущена без будь-якої модифікації. Єдине обмеження полягає в тому, що вона повинна підтримувати реальне фізичне обладнання.

Паравіртуалізація – це інший вид віртуалізації, який має деяку схожість з повною віртуалізацією. Цей метод теж використовує гіпервізор для розділення доступу до обладнання, але інтегрує код віртуалізації в саму операційну систему як показано на рисунку 1.3. Цей підхід дозволяє уникнути будь-якої перекомпіляції або перехоплення команд, оскільки операційна система сама бере участь в процесі віртуалізації.



Рисунок 1.7 Схема паравіртуалізації

Паравіртуалізація вимагає модифікації гостьової операційної системи, що є недоліком. Однак в цьому випадку забезпечується продуктивність, близька до продуктивності не віртуалізованої системи. Подібно до повної віртуалізації безліч різних операційних систем можуть підтримуватися одночасно [19].

Віртуалізація рівня операційної системи, використовує техніку, що відрізняється від тих способів, які були розглянуті раніше. У цьому варіанті віртуалізуються сервери, що запускаються в операційній системі. У цьому випадку операційна система одна і просто ізолюються один від іншого сервера, що працюють під її управлінням. Схему такого типу віртуалізації представлено на рисунку 1.8.



Рисунок 1.8 Схема віртуалізація рівня операційної системи

Віртуалізація рівня операційної системи вимагає внесення змін в ядро операційної системи, але зате дозволяє досягти вихідної її продуктивності.

1.5 Узагальнена модель керування якістю ІТ-сервісів

В даному розділі описується загальний вигляд моделі керування якістю ІТ-сервісів та вводяться основні позначення. В подальших розділах модель буде ускладнюватися або видозмінюватися.

Позначимо обчислювальні ресурси постачальника ІТ-сервісу як множину \mathfrak{R} , на базі яких він надає множину сервісів S . На обладнання, що надає ІТ-послуги надходить потік запитів користувачів Req , на які воно відповідає потоком відповідей $Resp$.

Перед системою керування рівнем якості надаваних сервісів ставиться задача задіяти ресурси R таким чином, щоб рівень якості кожного сервісу S максимально задовольняв оговореному рівню. При чому, система має враховувати множину збуджуючих впливів $Dist$ які впливають на ресурси \mathfrak{R} та сервіси S . Для вирішення поставленої задачі система керування рівнем якості сервісів має обчислювати рівень якості кожного із сервісів та базуючись на отриманий рівень якості сервісів генерувати необхідний множину керуючих впливів K на ресурси \mathfrak{R} .

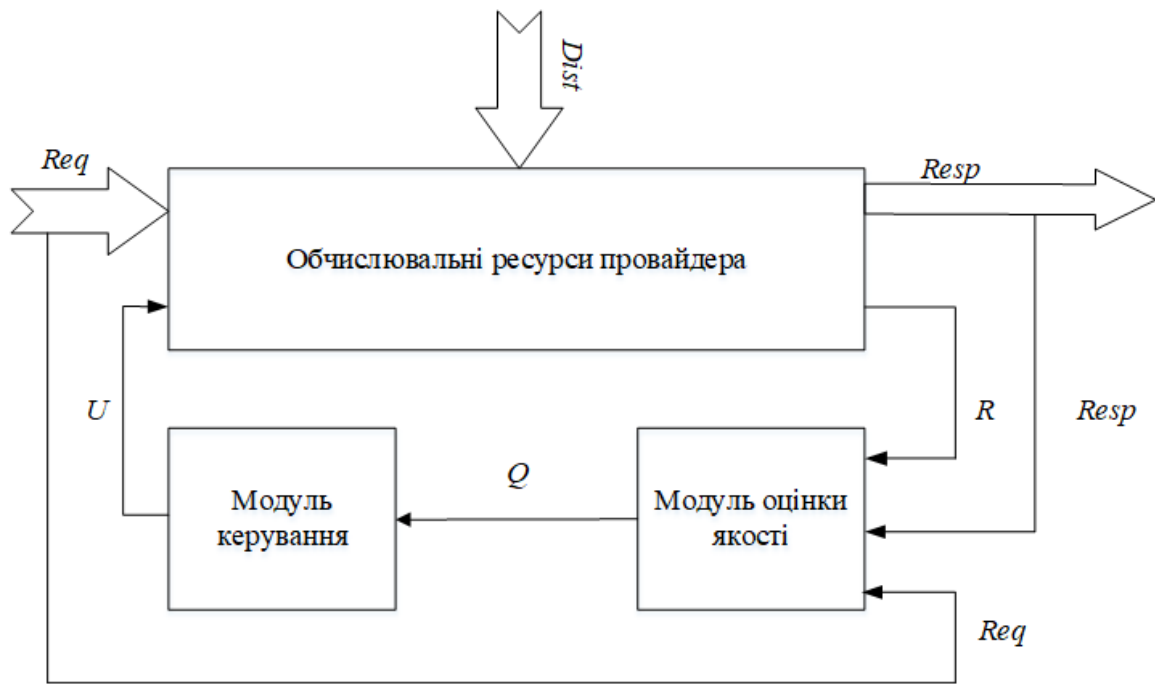


Рисунок 1.9 Узагальнена структура системи управління якістю ІТ-сервісу

На рисунку 1.9 схематично зображено модель управління якістю надаваних сервісів. Такий підхід дозволяє підтримувати сервіси S на заданому рівні якості та раціонально використовувати ресурси постачальника \mathcal{R} .

Для реалізації описаної системи управління, необхідно вирішити наступні завдання:

1. Визначити способи моніторингу обчислювальних ресурсів провайдерів сервісів.
2. Розробити моделі та методи розрахунку оцінки якості ІТ-сервісів на базі значень даних моніторингу обчислювальних ресурсів провайдерів сервісів.
3. Розробити моделі та методи автоматичного управління ресурсами провайдерів ІТ-сервісів для забезпечення узгодженого рівня якості ІТ-сервісів, що надаються провайдером.

Висновки до першого розділу

1. Проведено аналіз проблем забезпечення якості надаваних ІТ-сервісів, що дозволило визначити основні функції ІТ-підрозділів, що постачають дані сервіси: підтримка ІТ-інфраструктури; підтримка бізнес-додатків; підтримка

користувачів. Основною метою постачальників ІТ-сервісів є максимальна задоволеність користувачів ІТ-послуг при мінімальних затратах ІТ-ресурсів.

2. Аналіз проблем забезпечення якості надаваних провайдером ІТ-сервісів підкреслив доцільність раціонального використання ІТ-ресурсів. За обмеженого об'єму ІТ-ресурсів у постачальника ІТ-сервісів, постачальникам необхідно перерозподіляти ресурси між ІТ-послугами та визначати доцільність надання маловажливих послуг з високою якістю. Визначено три рівні аналізу якості ІТ-сервісів. Базуючись на аналізі показників цих трьох рівнів сервісів, ІТ-підрозділи визначають дії, які необхідно виконати для забезпечення якості надаваних ІТ-сервісів на узгодженому рівні.

4. Проаналізовано задачу визначення рівня задоволеності користувача якістю сервісів, що надаються і зведено її до задачі визначення залежності якості надаваних сервісів від значень окремих параметрів якості ІТ-сервісів. Це дає змогу автоматично визначати відповідність якості наданих ІТ-сервісів угоді про рівень послуг.

РОЗДІЛ 2 МОДЕЛІ І МЕТОДИ ОЦІНЮВАННЯ ТА УПРАВЛІННЯ РІВНЕМ ЯКІСТІ ІТ-ПОСЛУГ

2.1 Підхід до управління рівнем якості надаваних ІТ-послуг

Зараз при врегулюванні відносин між постачальниками та отримувачами ІТ-послуг превалює клієнт-орієнтований підхід. Цей підхід передбачає, що вся діяльність провайдера ІТ-послуг повинна будуватись таким чином, щоб забезпечити максимальну задоволеність користувача якістю ІТ-послуг, які надає провайдер цих послуг. В той же час, при розгляді аспектів надання ІТ-послуг користувачам сучасних корпоративних ІТ-інфраструктур відбувається зміна ролі ІТ-організацій, які перетворюються у посередника, що пропонує загальні та індивідуальні ІТ-послуги [30]. Цьому сприяє зміна орієнтирів ІТ-підрозділу, якій зараз витрачає біля 70% ІТ-бюджету на підтримку працездатності ІТ-інфраструктури, обслуговування систем та обладнання, а не на підтримку користувачів й підвищення їх задоволеності. Подібно ідеології бізнесу, згідно який бізнес перш за все повинен бути корисним користувачам, ІТ-організації необхідно будувати свою діяльність за принципом максимального задоволення інтересів користувачів.

Тому в роботі застосовано клієнт-орієнтований підхід до управління рівнем якості ІТ-послуг у ІТ-інфраструктурах корпоративного рівня. Основним критерієм оцінки якості надання ІТ-послуг ІТ-інфраструктурою при такому підході виступає рівень задоволеності користувачів якістю ІТ-послуг Ω .

Оскільки основним механізмом управління ІТ-інфраструктурою, який використовується у даній роботі задля підтримки якості послуг на узгодженому рівні є зміна обсягів її обчислювальних та комунікаційних ресурсів з множини \mathfrak{R} , що виділяються для надання послуг, то перш за все для кожної послуги необхідно визначити залежність Ω цією послугою від об'ємів ресурсів з множини \mathfrak{R} , які виділені для надання цієї послуги. Крім того необхідно враховувати, що на якість надання ІТ-послуг негативно впливають такі фактори,

як кількість запитів користувачів, несправності в ІТ-інфраструктурі, перенавантаження серверів або каналів зв'язку, тощо.

Позначимо як \mathcal{Z} множину всіх впливів на функціонування ІТ-інфраструктури та, відповідно, на якість послуг. Тоді

$$q: (\mathcal{R}, \mathcal{Z}) \rightarrow \Omega, \quad (2.1)$$

де q – функція залежності задоволеності користувача якістю послуг від обсягів ресурсів ІТ-інфраструктури з множини \mathcal{R} , що виділено для надання цих послуг та зовнішніх впливів на ІТ-інфраструктуру \mathcal{Z} .

Загальна задоволеність користувача якістю ІТ-послуг Ω визначається трьома показниками задоволеності:

$$F: (\mathcal{C}_{CSI}, \mathcal{C}_{CRM}, \mathcal{C}_{Serv}) \rightarrow \Omega, \quad (2.2)$$

де \mathcal{C}_{CSI} – індекс задоволеності користувачів (Customer Satisfaction Index), суб'єктивний фактор, якій визначається задоволеністю користувачів постачальником ІТ-послуг. Значення показника \mathcal{C}_{CSI} не розраховується, а визначається за методиками, які враховують психологічні та емоційні особливості людини, а також то, наскільки для споживача важливі конкретні ІТ-послуги. Зазвичай, вклад цього показника у Ω не перевищує 5-10%;

\mathcal{C}_{CRM} – рівень задоволеності користувача системою CRM управління відносинами з користувачами. Цей показник також не розраховується, а визначається системою CRM шляхом опитування користувачів. Зазвичай, вклад цього показника у Ω не перевищує 10-15%;

\mathcal{C}_{Serv} – інтегральний показник якості ІТ-послуг;

F – функціонал залежності задоволеності користувача якістю послуг Ω від значень показників задоволеності $\mathcal{C}_{CSI}, \mathcal{C}_{CRM}, \mathcal{C}_{Serv}$.

Інтегральний показник якості ІТ-послуг \mathfrak{C}_{Serv} складається з загального рівня задоволеності користувача усіма послугами, що надаються йому ІТ-інфраструктурою:

$$\mathcal{F}:(\mathfrak{C}_1, \mathfrak{C}_2, \dots, \mathfrak{C}_E) \rightarrow \mathfrak{C}_{Serv}, \quad (2.3)$$

де \mathfrak{C}_e – рівень задоволеності користувача якістю e -ї послугою, $e = \overline{1, \dots, E}$, а \mathcal{F} – функціонал залежності рівня задоволеності користувача від комплексної якості всіх послуг, що надаються йому ІТ-інфраструктурою.

Оскільки зміною кількості обсягів ресурсів з множини \mathfrak{R} не можна впливати на значення показників \mathfrak{C}_{CSI} та \mathfrak{C}_{CRM} , то залежність (2.2) при розробці методів та інформаційних технологій управління якістю ІТ-послуг у корпоративної ІТ-інфраструктурі зводиться до

$$\mathfrak{Q} = \mathfrak{C}_{Serv} = \mathcal{F}(\mathfrak{C}_1, \mathfrak{C}_2, \dots, \mathfrak{C}_E). \quad (2.4)$$

Для спрощення загальної моделі керування якістю ІТ-послуг та без втрати загальності будемо вважати ресурсами з множини \mathfrak{R} ІТ-інфраструктури обчислювальні ресурси серверів, на кожному з яких працює гіпервізор.

Нехай ІТ-інфраструктура містить N серверів, на кожному з яких запущено один гіпервізор. Позначимо множину гіпервізорів як \mathfrak{R}_H , тоді:

$$\begin{cases} \mathfrak{R}_H = \{H_\eta \mid \eta = \overline{1, \dots, N}\} \\ \mathfrak{R}_H \in \mathfrak{R} \end{cases} \quad (2.5)$$

Нехай кожен гіпервізор H_η , $\eta = \overline{1, \dots, N}$ створює множину віртуальних машин V_η , а кожна віртуальна машина надає лише одну ІТ-послугу з множини E :

$$V_\eta = \{V_{\eta\varpi_\eta e} \mid \varpi_\eta = \overline{1, \dots, \varsigma_\eta}, e \in [1, E]\}, \quad (2.6)$$

де ς_η – кількість віртуальних машин, роботу яких забезпечує гіпервізор H_η .

Позначимо \mathfrak{R}_e як ресурси що задіяні для надання e -го сервісу, \mathfrak{R}_{V_η} – ресурси, що використовує гіпервізор H_η , $\mathfrak{R}_{V_{\eta\varpi_\eta o}}$ – ресурси, що задіяні для роботи віртуальної машини $V_{\eta\varpi_\eta o}$, $e = \overline{1, \dots, E}$, $\eta = \overline{1, \dots, H}$, $\varpi_\eta = \overline{1, \dots, \varsigma_\eta}$, $o \in [1, O]$, а ресурси що не задіяні гіпервізорами, але використовуються для надання e -го сервісу як $\tilde{\mathfrak{R}}_e$. Тоді:

$$\begin{cases} \mathfrak{R} = \{\mathfrak{R}_e \mid e = \overline{1, \dots, E}\}; \\ \mathfrak{R}_e = \tilde{\mathfrak{R}}_e \cup \{\mathfrak{R}_{V_{\eta\varpi_\eta o}} \mid \eta \in [1, H], \varpi_\eta \in [1, \varsigma_\eta], o \in [1, O]\} \forall e = \overline{1, \dots, E}. \end{cases} \quad (2.7)$$

Позначимо як $\mathfrak{Z}_{\eta\varpi_\eta o}$, $\eta = \overline{1, \dots, H}$, $\varpi_\eta = \overline{1, \dots, \varsigma_\eta}$, $o \in [1, O]$ множину зовнішніх факторів, що впливають на роботу віртуальної машини $V_{\eta\varpi_\eta o}$, а як \mathfrak{Z}_e – зовнішні впливи на ІТ-інфраструктуру, тоді:

$$\begin{cases} \mathfrak{Z} = \{\mathfrak{Z}_{\eta\varpi_\eta o} \mid \varpi_\eta = \overline{1, \dots, \varsigma_\eta}, o \in [1, O]\}; \\ \mathfrak{Z} = \{\mathfrak{Z}_e \mid e = \overline{1, \dots, E}\}. \end{cases} \quad (2.8)$$

За такої умови, задачу управління ІТ-інфраструктурою з врахуванням (2.4) можна сформулювати таким чином. Для зміни значення показника \mathfrak{C}_{Serv} необхідно здійснювати керуючі впливи на обсяги ресурсів з множини \mathfrak{R} таким чином, щоб досягти бажаного рівня \mathfrak{Q} , враховуючи \mathfrak{Z} .

Для вирішення цієї задачі та розробки відповідного метода керування перш за все необхідно встановити вид функціональної залежності q_e :

$$q_e : (\mathfrak{R}_e, \mathfrak{Z}_e) \rightarrow \mathfrak{C}_e. \quad (2.9)$$

для всіх $e = \overline{1, \dots, E}$.

Враховуючи (2.7) та (2.8) залежність (2.9) можна визначити як

$$q_e : \left(\mathcal{Z}_e, \tilde{\mathcal{X}}_e, \left\{ \mathcal{R}_{V_{\eta\varpi\eta^o}} \mid \eta \in [1, N], \varpi_\eta \in [1, \zeta_\eta], o \in [1, O] \right\} \right) \rightarrow \mathfrak{C}_e \quad (2.10)$$

для всіх $e = \overline{1, \dots, E}$.

На рисунку 2.1 показано, що вирішення задачі (2.10) здійснюється шляхом поетапного зведення метрик.

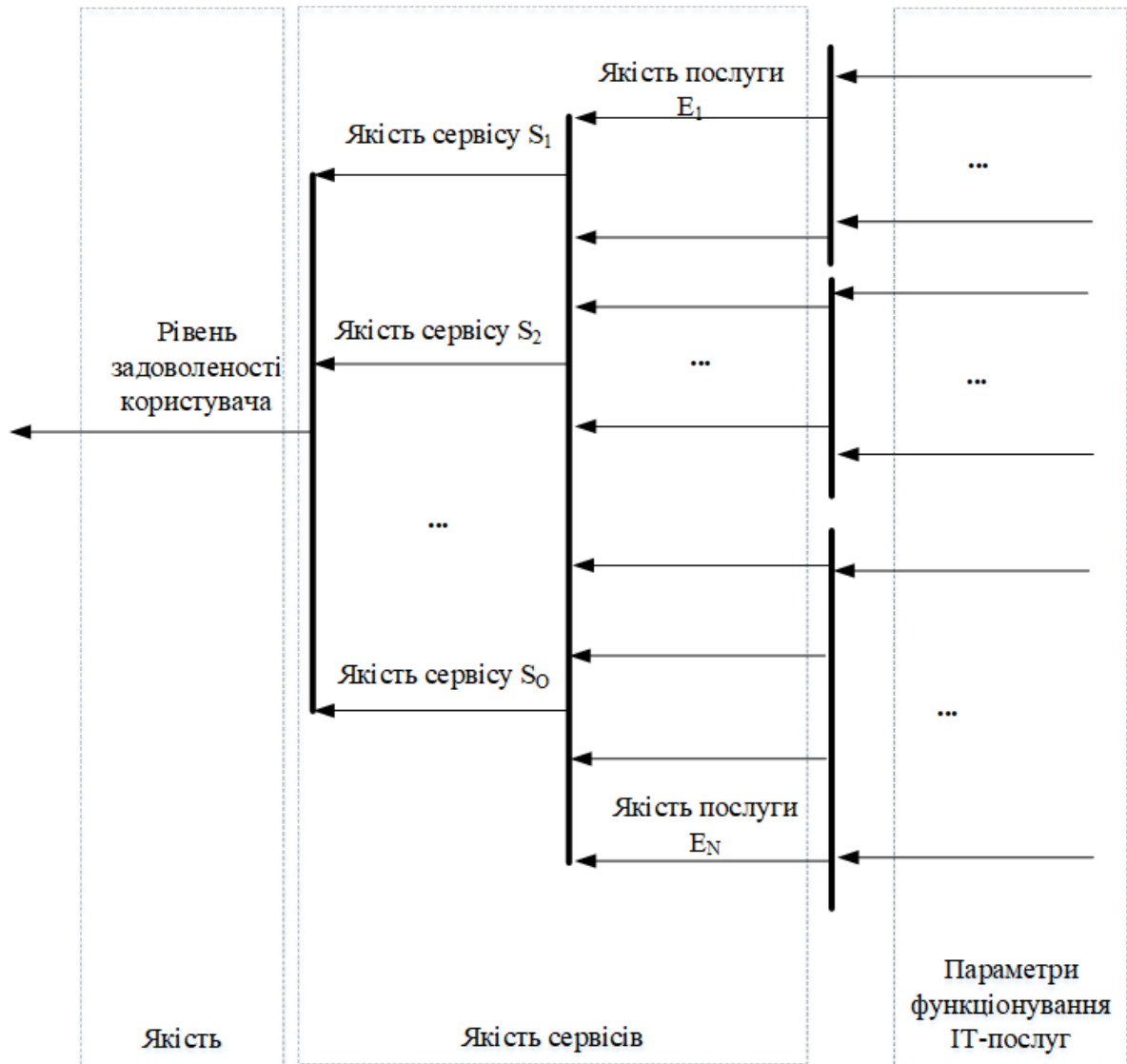


Рисунок 2.1 Зведення метрик для визначення значення показника \mathfrak{C}_{serv}

Якщо залежність (2.10) встановлена, то СУІ має змогу визначати поточне значення Ω . Але для впливу ІТ-інфраструктуру з метою підтримки якості ІТ-послуг на узгодженому рівні необхідно визначити на скільки необхідно змінити об'єми ресурсів, що виділені ІТ-інфраструктурою для надання кожної окремої послуги, тобто знайти залежності

$$p_e : (\mathfrak{C}_e, \mathfrak{Z}_e) \rightarrow \mathfrak{R}_e \cup \mathfrak{R}_e \quad (2.11)$$

Обидві задачі (2.10) та (2.11) вирішуються далі у даній роботі.

Позначимо як $P_{oe} = \{p_{oe\rho} \mid \rho = \overline{1, \dots, P_e}, e = \overline{1, \dots, E_o}\}$ множину параметрів, що характеризують e -у послугу для всіх $o = \overline{1, \dots, O}$, $P_{\eta\varpi_\eta o} = \{p_{\eta\varpi_\eta o\rho} \mid \rho = \overline{1, \dots, P_{\eta\varpi_\eta o}}\}$ множину параметрів, що характеризують віртуальну машину $V_{\eta\varpi_\eta o}$, а кількісні характеристики кожного параметру як $P'_{oe} = \{p'_{oe\rho} \mid \rho = \overline{1, \dots, P_{oe}}\}$ та $P'_{\eta\varpi_\eta o} = \{p'_{\eta\varpi_\eta o\rho} \mid \rho = \overline{1, \dots, P_{\eta\varpi_\eta o}}\}$ відповідно, де $P_{\eta\varpi_\eta o}$ – кількість параметрів, що характеризують віртуальну машину $V_{\eta\varpi_\eta o}$, P_{oe} – кількість параметрів, що характеризують послугу E_{oe}

Таким чином:

$$\forall (p_{\eta\varpi_\eta o\rho} \in P_{\eta\varpi_\eta o}) \exists p'_{\eta\varpi_\eta o\rho}. \quad (2.4)$$

для всіх $\eta \in [1, H], \varpi_\eta \in [1, \varsigma_\eta], o \in [1, O], \rho \in [1, P_{\eta\varpi_\eta o}]$.

Узагальнимо множини параметрів:

$$\begin{aligned}
P &= \bigcup_{\eta, \varpi_\eta, o} P_{\eta \varpi_\eta o}, \\
P' &= \bigcup_{\eta, \varpi_\eta, o} P'_{\eta \varpi_\eta o}, \\
\tilde{P} &= \bigcup_{o, e} P_{oe}, \\
\tilde{P}' &= \bigcup_{o, e} P'_{oe}.
\end{aligned} \tag{2.5}$$

Позначимо оцінку якості o -го сервісу, який надається віртуальною $\varpi_\eta o$ -ю машиною η -го гіпервізору як $Q_{\eta \varpi_\eta o}$, де $o \in [1, O]$, $\eta \in [1, H]$, $\varpi_\eta \in [1, \Pi_\eta]$. Задачею модулю оцінки якості полягає у визначенні рівня якості сервісів множини S , для кожної віртуальної машини $V_{\eta \varpi_\eta o} \in \bigcup_{\eta=1}^H V_\eta$ на базі значень множини параметрів $P'_{\eta \varpi_\eta o}$ обчислити рівень якості $Q_{\eta \varpi_\eta o}$ сервісу, що надає віртуальна машина $V_{\eta \varpi_\eta o}$. Позначимо для o -го сервісу функцію залежності значення $Q_{\eta \varpi_\eta o}$ від множини значень параметрів $P'_{\eta \varpi_\eta o}$ як Θ_o , тоді:

$$\Theta_o : \left(p'_{\eta \varpi_\eta o 1}, p'_{\eta \varpi_\eta o 2}, \dots, p'_{\eta \varpi_\eta o P_{\eta \varpi_\eta o}} \right) \rightarrow Q_{\eta \varpi_\eta o}. \tag{2.6}$$

для всіх $\eta \in [1, H]$, $\varpi_\eta \in [1, \varsigma_\eta]$, $o \in [1, O]$, $\rho \in [1, P_{\eta \varpi_\eta o}]$.

Варто зауважити, що така функція індивідуальна для кожного окремого сервісу. Для її знаходження необхідно визначити один або декілька показників якості сервісу та знайти залежність між значеннями параметрів віртуальної машини, що надає сервіс та значеннями цих показників.

2.2 Метод оцінки якості ІТ- сервісів на основі простих нейромережевих класифікаторів

2.2.1 Розробка методу оцінки якості ІТ-послуг на основі простих нейромережевих класифікаторів

Для спрощення припустимо, що кожен сервіс має один показник якості. Далі буде розглянуто методи зведення декількох показників якості надаваного сервісу до єдиного інтегрального показника якості. Таким чином, запропонований метод дозволяє обчислювати рівні якості Q надаваних сервісів на основі аналізу ресурсів постачальника сервісу \mathfrak{R} та значень параметрів P' .

Так як розглядається випадок коли одному сервісу виділяється окрема віртуальна машина, то для вирішення даної задачі для кожного окремого сервісу немає потреби враховувати ресурси постачальника, які підтримують роботу інших сервісів. Виходячи з цього, задача оцінки якості всіх сервісів, що належать до множини S зводиться до $\sum_{\eta=1}^N \zeta_{\eta}$ окремих задач. При цьому, для кожної з цих задач будуть враховуватися лише значення параметрів окремо взятої віртуальної машини.

Розглянемо віртуальну машину $V \in \{V_{\eta} \mid \eta = \overline{1, \dots, N}\}$ яка надає сервіс $S \in S$, нехай дана машина має множину параметрів $P \in P$, що впливають на якість сервісу S .

Тоді задача оцінки якості сервісу S зводиться до задачі пошуку функції залежності:

$$\Theta: \left(p_1, p_2, \dots, p_{|P|} \right) \rightarrow Q, \quad (2.7)$$

де $p_1, p_2, \dots, p_{|P|}$ – відповідні значення параметрів множини P . Цю задачу буде вирішувати нейромережевий класифікатор. На його вході буде $|P|$ вхідних нейронів, по одному на кожен параметр віртуальної машини, а на виході він буда надавати значення рівня якості сервісу S .

Для навчання такої нейронної мережі необхідно створити навчальну вибірку. Для цього необхідно з високою точністю вказати значення рівня якості сервісу S при тих чи інших значеннях параметрів множини P . Не для кожного сервісу є можливість чітко визначати такі значення. Тому запропоновано використовувати три класи якості: відмінна, задовільна, незадовільна. Клас відмінно позначається літерою A , задовільно та незадовільно – B та C відповідно [20].

Тоді структура модулю оцінки якості матиме вигляд як на рис. 2.2.

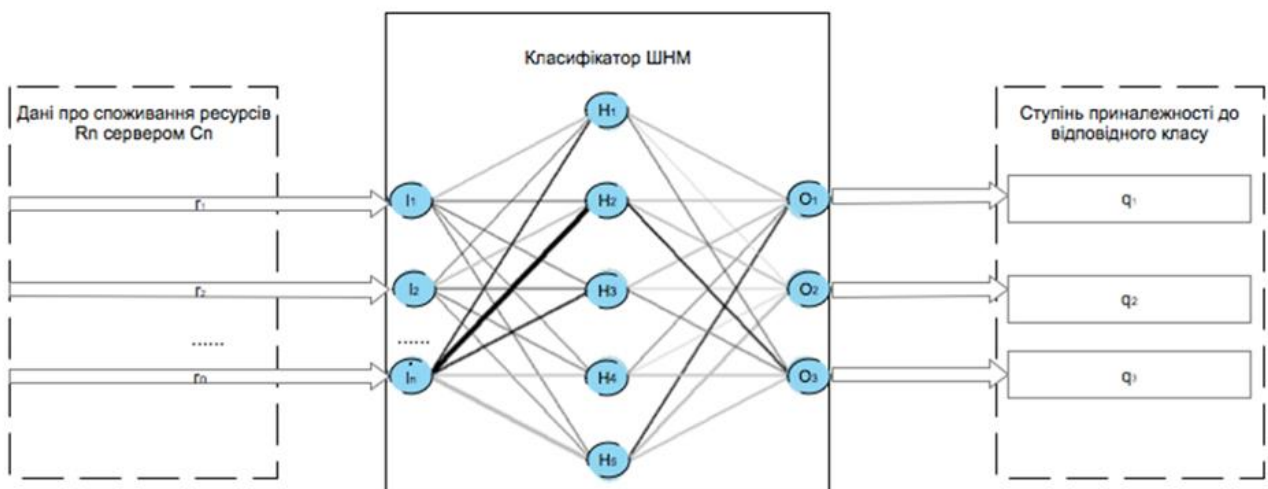


Рисунок 2.2 Схема класифікатора ШНМ, який використовується в методі розрахунку рівня якості сервісів

На виході такого нейромережевого класифікатора отримуємо ступені належності рівня якості сервісу S до класів A , B та C .

Такий підхід дасть змогу легше налаштовувати модулі оцінки якості у системі автоматичного керування сервісами, адже людині легше оперувати

лінгвістичними значеннями, аніж чітко визначати числове значення рівня якості сервісу. Також, в подальшому, це дасть можливість з більшою легкістю налаштовувати правила для роботи модулю керування.

З точки зору машинного навчання, нейронна мережа є окремим випадком методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо. З математичної точки зору, навчання нейронних мереж – це багатопараметрична задача нелінійної оптимізації. З точки зору кібернетики, нейронна мережа використовується в задачах адаптивного управління і як алгоритми для робототехніки. З точки зору розвитку обчислювальної техніки та програмування, нейронна мережа – спосіб вирішення проблеми ефективного паралелізму. А з точки зору штучного інтелекту, штучні нейронні мережі напрямком в структурному підході з вивчення можливості побудови (моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів [21].

Нейронні мережі не програмуються в звичайному розумінні цього слова, вони навчаються. Можливість навчання – одна з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що у разі успішного навчання мережа зможе повернути правильний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних та/або зашумлених, частково перекручених даних.

Було розглянуто декілька типів штучних нейронних мереж – перцептрон [21-23], нечіткі нейронні мережі [22-24], мережі Кохонена та мережу радіально-базисних функцій [21, 24].

Мережі РБФ мають ряд переваг перед розглянутими багат шаровими мережами прямого поширення. По-перше, вони моделюють довільну нелінійну функцію за допомогою всього одного проміжного шару, тим самим, позбавляючи розробника необхідності вирішувати питання про кількість шарів. По-друге, параметри лінійної комбінації в вихідному шарі можна повністю

оптимізувати за допомогою добре відомих методів лінійної оптимізації, які працюють швидко і не відчують труднощів з локальними мінімумами, так заважають при навчанні з використанням алгоритму зворотного поширення помилки. Тому мережа РБФ навчається швидше, ніж з використанням алгоритму зворотного поширення помилки [25].

Мережі РБФ мають два основних недоліка: вони виходять громіздкими при великій розмірності вектору входів та володіють поганими екстраполяційними властивостями. Перший недолік не стає на заваді вибору даного типу нейронних мереж, бо вхідний вектор в нашому випадку не є великим. Та враховуючи специфіку поставленої в даній роботі перед нейронною мережею задачі – побудова класифікатора при можливій неповній навчальній вибірці, даний тип нейронних мереж не було вибрано.

Мережа Кохонена (SOM) складається з одного шару ваг, що налаштовуються і функціонують, виходячи зі стратегії збудження тільки одного нейрону та придушення інших виходів шару. Мережа Кохонена здійснює класифікацію вхідних векторів в групи схожих, підлаштовуючи ваги таким чином, що вхідні образи, що належать одному класу, будуть активувати один і той же вихідний нейрон. Одне з найважливіших властивостей навчальної мережі Кохонена – здатність до узагальнення. Вектор кожного з нейронів мережі замінює групу відповідних йому векторів, що класифікують.

Нейронні мережі такого типу не підходять для виконання поставленої задачі, так як розмірність шару ваг, що налаштовується буде дуже великою. Також при можливій неповній навчальній вибірці дана мережа може дати непередбачувані результати.

Зважаючи на всі переваги та недоліки, в даному методі оцінки якості роботи сервісів віртуальних серверів запропоновано використовувати такий тип штучних нейронних мереж, як перцептрон [22]. В подальшому тексті перцептрон буде називатися штучною нейронною мережею (ШНМ).

2.2.2 Вибір методу навчання штучної нейронної мережі

Серед методів навчання доцільно виділити наступні методи: метод зворотного поширення помилки, метод сполучених градієнтів, метод Левенберга-Маркара, метод градієнтного спуску з адаптивним навчанням, метод Дельта-дельта з межею.

Метод зворотного поширення помилки – це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багатoshарового перцептрону і отримання бажаного виходу [22].

Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Для можливості застосування цього методу передавальна функція нейронів повинна бути диференційована.

Алгоритм зворотного поширення помилки застосовується для багатoshарового перцептрону. Через особливості обчислення поправок – обчислювати поправку для вузлів останнього рівня і висловлювати поправку для вузла нижчого рівня через поправки вищого – алгоритм називається алгоритмом зворотного поширення помилки. На вхід алгоритму, крім зазначених параметрів, потрібно також подавати в якомусь форматі структуру мережі. На практиці дуже хороші результати показують мережі досить простої структури, що складаються з двох рівнів нейронів – прихованого рівня і нейронів-виходів; кожен вхід мережі з'єднаний з усіма прихованими нейронами, а результат роботи кожного прихованого нейрона подається на вхід кожного з нейронів-виходів. У такому випадку досить подавати на вхід кількість нейронів прихованого рівня.

Незважаючи на численні успішні застосування зворотного поширення найбільше неприємностей приносить невизначено довгий процес навчання. У складних завданнях мережу може і взагалі не навчитися. Причини можуть бути наступні:

- параліч мережі. У процесі навчання мережі значення ваг можуть в результаті корекції стати дуже великими величинами. Більшість

нейронів будуть функціонувати в області, де похідна стискає функції дуже мала. Процес навчання може практично завмерти.

- локальні мінімуми. Зворотне поширення використовує різновид градієнтного спуску – спуск вниз по поверхні помилки - безперервно підлаштовуючи ваги в напрямку до мінімуму. Мережа може потрапити в локальний мінімум, коли поруч є набагато більш глибокий мінімум.
- розмір кроку. Корекції ваг передбачаються нескінченно малими. Це нездійсненно на практиці, так як веде до нескінченного часу навчання. Однак розмір кроку повинен братися кінцевим.

Слід також відзначити можливість перенавчання мережі, що є скоріше результатом помилкового проектування її топології. При занадто великій кількості нейронів втрачається властивість мережі узагальнювати інформацію. Весь набір образів, наданих до навчання, буде вивчено мережею, але будь-які інші образи, навіть дуже схожі, можуть бути класифіковані невірно.

Існують сучасні алгоритми другого порядку, такі як метод сполучених градієнтів і метод Левенберга-Маркара, які на багатьох завданнях працюють істотно швидше (іноді на порядок). Розроблено також евристичні модифікації цього алгоритму, добре працюють для певних класів задач, – швидке поширення і Дельта-дельта з межею [21, 23].

Метод спряжених градієнтів – метод знаходження локального мінімуму функції на основі інформації про її значення і її градієнту. У разі квадратичної функції в R_n мінімум знаходиться за n кроків.

Метод спряжених градієнтів є методом першого порядку, в той же час швидкість його збіжності квадратична. Цим він вигідно відрізняється від звичайних градієнтних методів. Наприклад, метод найшвидшого спуску і метод координатного спуску для квадратичної функції сходяться лише в межі, в той час як метод сполучених градієнтів оптимізує квадратичну функцію за кінцеве число ітерацій. При оптимізації функцій загального вигляду, метод сполучених напрямків сходиться в 4-5 разів швидше методу найшвидшого спуску. При

цьому, на відміну від методів другого порядку, не потрібно трудомістких обчислень друге приватних похідних [24].

Метод градієнтного спуску з урахуванням моментів. Ідея методу полягає в додаванні до величини корекції ваги значення пропорційного величині попереднього зміни цього ж вагового коефіцієнту.

$$\Delta w = -\eta \cdot \frac{\partial E}{\partial w} + \alpha \cdot \Delta w(t-1), \quad (2.8)$$

де $\Delta w(j)$ – значення корекції ваги на j -му кроці, η – довжина кроку, $E(w)$ – функція помилки, α – коефіцієнт інерції.

Метод Левенберга-Маркара. Даний метод вважається найшвидшим і надійним алгоритмом навчання [24]. Однак його застосування пов'язано з певними обмеженнями:

- мережі з одним виходом – метод Левенберга-Маркара можна застосовувати тільки для мереж з одним вихідним елементом;
- невеликі мережі – метод Левенберга-Маркара вимагає пам'яті, пропорційних квадрату числа ваг в мережі; фактично це обмеження не дозволяє використовувати метод для мереж великого розміру (близько тисячі і більше ваг);
- ефективне значення функція помилок – метод Левенберга-Маркара застосуємо тільки для середньоквадратичної функції помилок; якщо зазначений для мережі інший вид функції помилок, то цей вибір буде проігнорований при навчанні методом Левенберга-Маркара; тому цей метод зазвичай підходить тільки для регресійних мереж.

Метод Левенберга-Маркара передбачає, що функція, моделюється нейронною мережею, є лінійної. В такому припущенні мінімум визначається за один крок обчислень. Потім знайдений мінімум перевіряється, і якщо помилка зменшилася, ваг присвоюються нові значення. Вся процедура послідовно повторюється. Оскільки припущення про лінійність, взагалі кажучи, не

виправдано, могло б вийти так, що довелося б перевіряти точки, що лежать далеко від поточної точки. У методі Левенберга-Маркара розташування нової точки є результат компромісу між просуванням у напрямку найшвидшого спуску і описаного вище стрибка. Успішні кроки приймаються, і баланс зміщується в бік припущення лінійності (яке приблизно вірно в околиці точки мінімуму). Невдалі кроки відкидаються, і алгоритм йде більш обережно вниз по схилу. Таким чином, алгоритм Левенберга-Маркара весь час змінює схему дії і може працювати дуже швидко [21, 22].

Алгоритм Левенберга-Маркара спеціально розроблений так, щоб мінімізувати середньоквадратичне функцію помилок за допомогою формули, яка (частково) передбачає, що функція, яку моделює мережу, є лінійної. Поблизу точки мінімуму це припущення виконується з великою точністю, так що алгоритм може просуватися дуже швидко. Далеко від мінімуму це припущення може бути неправильним. Тому метод Левенберга-Маркара знаходить компроміс між лінійної моделлю і градієнтним спуском. Крок робиться тільки в тому випадку, якщо він зменшує помилку, і там, де це необхідно, для забезпечення просування використовується градієнтний спуск з досить малим кроком.

Швидке поширення (метод градієнтного спуску з адаптивним навчанням). У методі швидкого поширення виробляється пакетна обробка даних. У той час як в методі зворотного поширення ваги мережі коригуються після обробки кожного чергового спостереження, в методі швидкого поширення обчислюється усереднений градієнт поверхні помилок по всьому навчальній множині, і ваги коректуються один раз в кінці кожної епохи [26].

Метод швидкого поширення діє в припущенні, що поверхня помилок є локально квадратичної. Якщо це так, то точка мінімуму на ній знаходиться всього через одну-дві доби. У загальному випадку таке припущення невірне, але навіть якщо воно виконується лише приблизно, алгоритм все одно дуже швидко сходиться до мінімуму.

У цьому припущенні алгоритм швидкого поширення працює так:

- на першій епосі ваги коректуються за тим же правилом, що і в методі зворотного поширення, виходячи з локального градієнта і коефіцієнта швидкості навчання;
- на наступних епохах алгоритм використовує припущення про квадратичності для більш швидкого просування до точки мінімуму.

Вихідні формули методу швидкого поширення мають ряд обчислювальних недоліків. По-перше, якщо поверхня помилок не є увігнутою, алгоритм може піти в хибному напрямку. Далі, якщо вектор градієнта не змінюється або змінюється мало, крок алгоритму може виявитися дуже великим і навіть нескінченним. Нарешті, якщо по ходу зустрівся нульовий градієнт, зміна ваг взагалі припиниться [24].

Метод швидкого поширення обробляє дані в пакетному режимі: градієнт помилки обчислюється як сума градієнтів помилок за всіма навчальним спостереженнями [23].

На першій епосі алгоритм швидкого поширення коригує ваги так само, як і алгоритм зворотного поширення. Потім зміни ваг визначаються за формулою швидкого поширення:

$$\Delta w(t) = \frac{s(t)}{s(t-1) - s(t)} \cdot \Delta w(t-1). \quad (2.9)$$

Дана формула чисельно нестійка при $s(t)$ близькому, рівному, або більшому, ніж $s(t-1)$. Оскільки $s(t)$ знаходиться після просування у напрямку градієнта, це може статися, тільки якщо нахил поверхні став постійним або стає крутіше (тобто поверхня не є увігнутою).

У таких випадках ваги коректуються за формулою:

$$\Delta w(t) = \alpha \cdot \Delta w(t-1), \quad (2.10)$$

де α – коефіцієнт прискорення.

Якщо градієнт стає рівним нулю, то приращення (дельта) ваги також робиться рівним нулю і, за наведеною вище формулою, так нулем і залишається, навіть якщо градієнт потім зміниться. Звичайний спосіб боротьби з такою трудністю полягає в додаванні до обчислених вище змін ваг маленького коефіцієнта. Однак це може привести до чисельної нестійкості. Якщо градієнт був нулем, а потім стає істотно відмінним від нуля, то коригування відповідної ваги знову робиться як для негативного градієнта [23].

Метод Дельта-дельта з межею (метод градієнтного спуску з урахуванням моментів і адаптивним навчанням). У деяких випадках він виявляється ефективніше, хоча в більшій мірі, ніж метод зворотного поширення, схильний застрягати в локальних мінімумах. На відміну від методу зворотного поширення, цей метод зазвичай досить стійкий [22].

У методі Дельта-дельта з межею швидкості навчання для окремих ваг коректуються на кожній епосі таким чином, щоб дотримувалися наступні важливі евристичні вимоги:

- якщо похідна зберігає знак на кількох послідовних ітераціях, то швидкість навчання збільшується (поверхню помилок має малу кривизну, тому однаковим просування відповідного приблизно однакові зниження рівня);
- якщо знак похідної на протязі декількох останніх ітерацій щоразу змінювався на протилежний, то швидкість навчання значно зменшується (якщо цього не зробити, алгоритм може почати осцилювати навколо точок з великою кривизною).

Щоб задовольнити цим умовам, в методі Дельта-дельта з межею задається початкова швидкість навчання, яка використовується для всіх ваг на першій епосі, коефіцієнт прискорення, який додається до швидкостей навчання, коли похідні не змінюють знаки, і коефіцієнт уповільнення, на який множаться швидкості навчання в разі, коли похідна змінює знак. Застосування лінійного росту і експоненціального убавання для швидкостей навчання надає алгоритму велику стійкість.

Однак описана схема може погано працювати на поверхнях помилок, перекручених перешкодами, де при вираженому загальному що знижується рельєфі похідні можуть різко змінювати знак. Тому при реалізації алгоритму для збільшення або зменшення швидкості навчання береться згладжений варіант похідною.

Ваги коректуються за тими ж формулами, що і в методі зворотного поширення, з тією різницею, що коефіцієнт інерції не використовується, а любий каприз має свою власну, незалежну від часу швидкість навчання.

На початку всім швидкостям навчання присвоюються однакові стартові значення; потім на кожній епосі вони коригуються [23].

Серед методів, що були розглянуті вище для подальшого дослідження було вибрано наступні: метод зворотного поширення помилки, метод спряжених градієнтів, метод градієнтного спуску з адаптивним навчанням, метод градієнтного спуску з урахуванням моментів і адаптивним навчанням.

Метод зворотного поширення помилки. На рисунку 2.2 зображено графік залежності похибки нейронної мережі від ітерації для навчальної та тестової вибірок. Вісь ординат відповідає похибці нейронної мережі, що проходить навчання, вісь абсцис – пройдена кількість ітерацій під час навчання. Чорним кольором зображено залежність для тестової вибірки, червоним – для навчальної вибірки.

Для навчання ШНМ були використані дані, які були отримані експериментальним шляхом. Ці дані включають в себе значення поточних параметрів $p_1, p_2, \dots, p_{|P|}$ та відповідної їм значення рівня якості сервісу S . Після навчання нейронна мережа має самостійно визначати поточну якість сервісу S . Рівень якості, яку визначає ШНМ, ґрунтуючись на значеннях поточних ресурсів $p_1, p_2, \dots, p_{|P|}$ має мати якомога менше значення похибки.

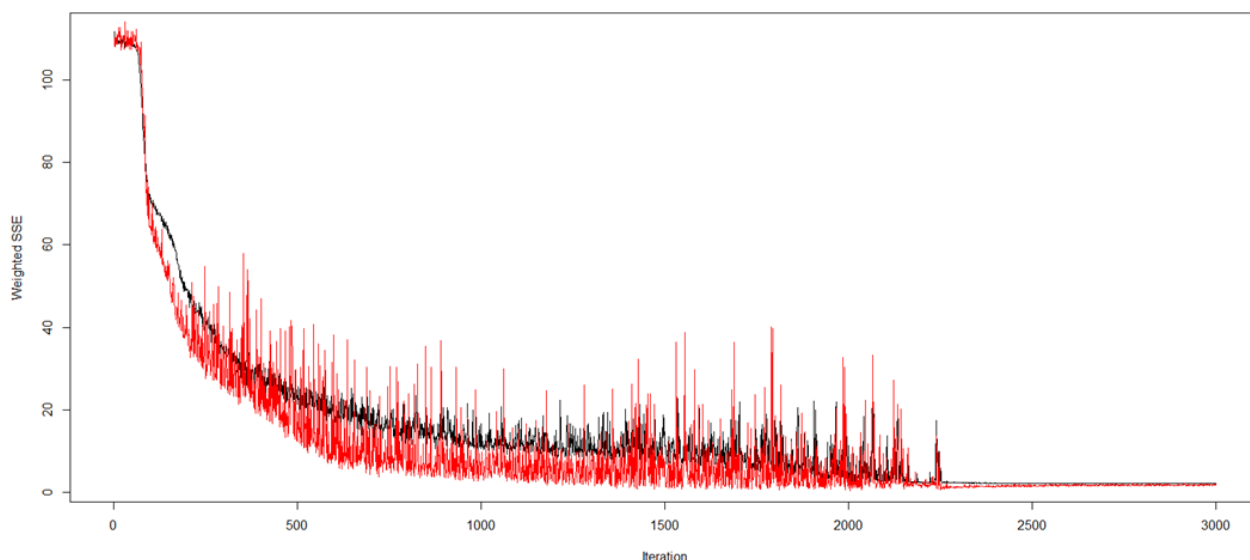


Рисунок 2.3 Метод зворотного поширення помилки

Як видно з рисунку 2.3 ШНМ, яка навчена методом зворотного зв'язку навчається 2200 ітерацій та похибка цієї нейромережі близька до нуля.

Графік залежності похибки нейронної мережі від ітерації для навчальної та тестової вибірок ШНМ, яка навчена методом спряжених градієнтів, зображено на рисунку 2.3. Похибка цієї нейронної мережі приблизна рівна нулю, та кількість ітерацій за яку ШНМ навчилася вірно визначати рівень якості сервісу значно менше за 1000.

Даний метод кращий за метод зворотного поширення помилки.

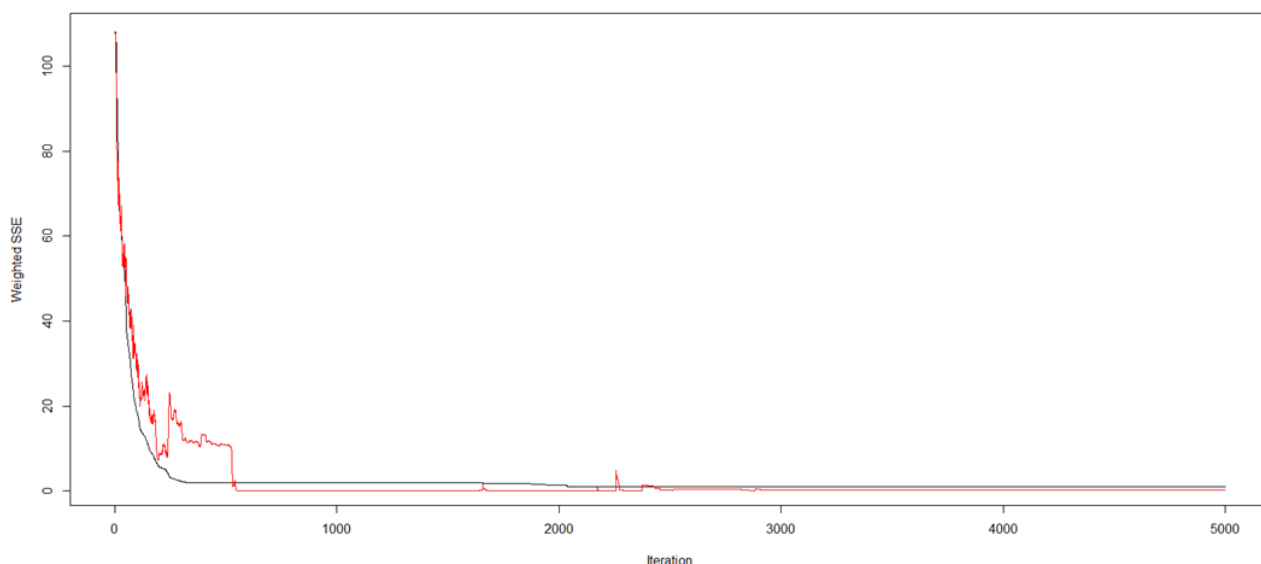


Рисунок 2.4 Метод спряжених градієнтів

На рисунку 2.4 зображено графік залежності похибки нейронної мережі від ітерації для навчальної та тестової вибірок ШНМ, яка навчена методом градієнтного спуску з адаптивним навчанням. Як видно з рисунку 2.4 даний метод значно гірший методу спряжених градієнтів та зворотного поширення помилки. Похибка ШНМ та кількість ітерацій набагато вищі за очікувану.

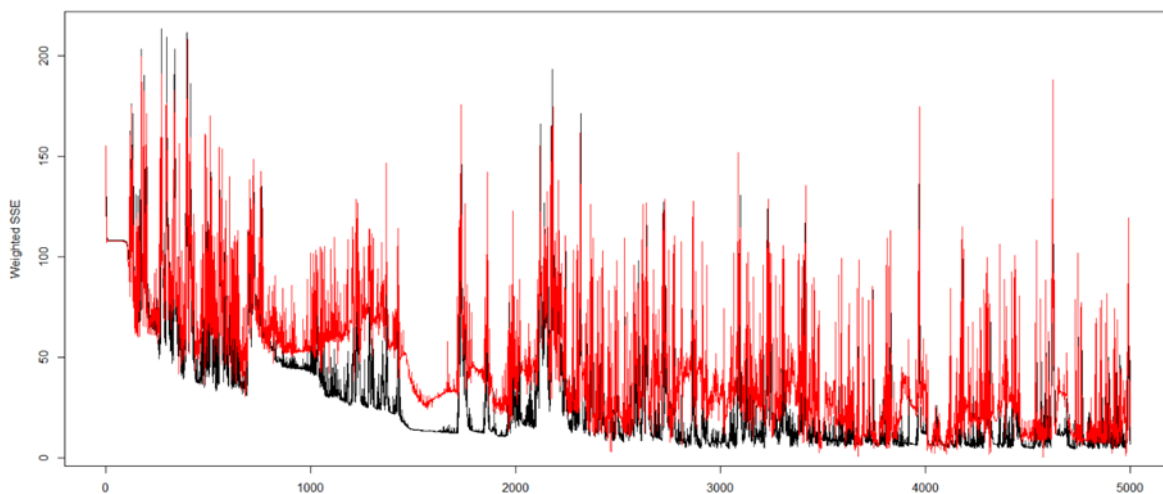


Рисунок 2.5 Метод градієнтного спуску з адаптивним навчанням

На рисунку 2.5 представлено графік залежності похибки нейронної мережі від ітерації для навчальної та тестової вибірок, яка навчена визначати рівень якості сервісу методом градієнтного спуску з урахуванням моментів і

адаптивним навчанням. Розглянутий метод навчає мережу правильно визначати якість сервісу за невелику кількість ітерацій, але з похибкою, яка більше 0.

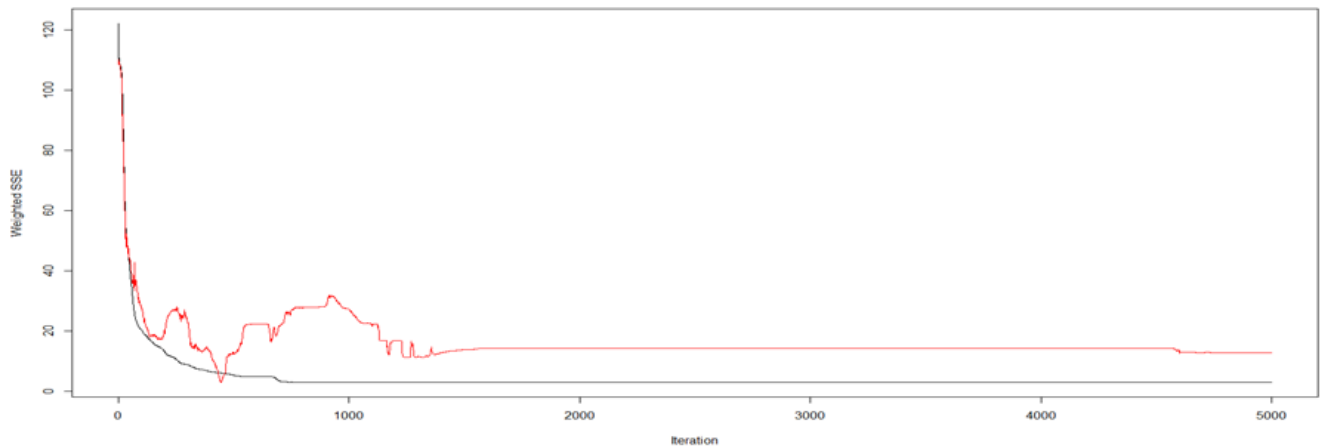


Рисунок 2.6 Метод градієнтного спуску з урахуванням моментів і адаптивним навчанням

З розглянутих методів навчання штучної нейронної мережі було вибрано метод спряжених градієнтів. Обраний метод забезпечує невелику похибку та невелику кількість ітерацій за яких ШНМ навчиться вірно визначати рівень якості сервісу.

2.2.3 Метод оцінки якості роботи ІТ-сервісу

Процеси всередині ІТ-інфраструктури, що надає сервіси, не завжди описуються або повністю оформлені, а в деяких випадках не вивчаються взагалі. Розроблений метод використовує нейронні мережі для оцінки якості функціонування, тому що нейронні мережі здатні ідентифікувати складні зв'язки між вхідними даними та вихідними даними та виконувати узагальнення. Це означає, що у випадку успішного навчання нейронна мережа зможе повернути правильний результат на основі даних, які бракувало в наборі тренувань, або на основі неповних або шумних, частково пошкоджених даних. ІТ-інфраструктура розвивається і змінюється з часом. Тому нейронна мережа, що навчається за застарілою інформацією, дасть неправильну оцінку. Зі зміною процесів

всередині інфраструктури не тільки коефіцієнти ваги нейронної мережі, але й структура нейронної мережі, можливо, також доведеться змінити. Тому запропонований метод використовує структурну підготовку нейронних мереж. Це дозволяє адаптуватися до динамічних змін в ІТ-інфраструктурі.

Запропонований алгоритм оцінки рівня якості роботи сервісів складається з кількох послідовних етапів:

1. Визначення параметрів віртуальних машин, що впливають на роботу сервісу. На цьому етапі визначається, що впливає на обраний сервіс.

2. Моніторинг. На цьому етапі відбувається збір інформації про функціонування елементів ІТ-інфраструктури, збираються значення параметрів віртуальних машин та створюється база знань, яка буде використовуватися як вхідна інформація для навчання нейронних мереж.

3. Тренінг. На цьому етапі визначаються структура та вагові коефіцієнти нейронних мереж.

4. Оцінка рівня якості роботи сервісу. На цьому етапі оцінюється якість роботи сервісу на основі даних, зібраних на кроці 2.

2.3 Структурна оптимізація нейромережевого класифікатора

На сьогоднішній день одиничні нейронні мережі широко використовуються для вирішення різних проблем. У зв'язку з цим дуже важливим є питання розробки алгоритму навчання, який дозволить динамічно оптимізувати структуру нейронних мереж. Існування такого методу дозволить досліднику отримати структуру нейронної мережі, яка буде кращим чином відповідати домену та доступними вхідними даними швидко.

В даному розділі для простоти подачі розглянуто загальний алгоритм структурного навчання нейронних мереж, без прив'язки до задач оцінки якості ІТ-сервісів.

Архітектура багат шарової нейронної мережі являє собою ациклічний граф G , представлений як сукупність вузлів, які згруповані в кластери за шарами та з'єднання між ними. Приклад такого представлення нейронної мережі

показано на рисунку 2.7. Кожен вузол $Node_i^j$ являє собою елементарний нейрон або елемент більш складної архітектури, наприклад, невелика нейронна мережа (рис. 2.8).

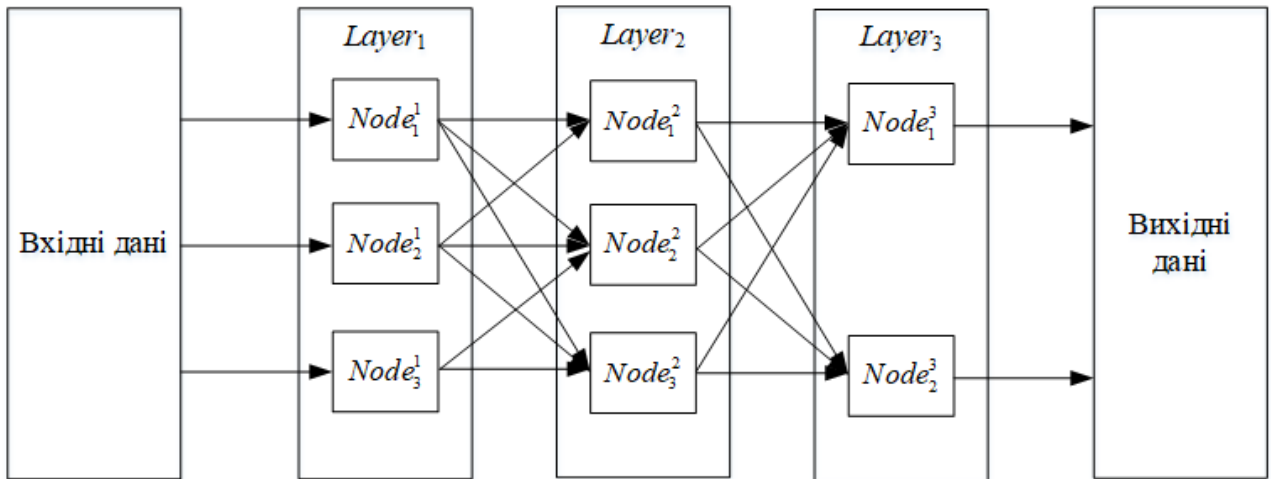


Рисунок 2.7 Графічна модель нейронної мережі

Запропонований алгоритм структурної підготовки мережі дозволяє:

- отримати загальну структуру нейронної мережі;
- визначити кількість нейронів у шарах нейронної мережі;
- видаляти незначні зв'язки між нейронами (синапсами);
- прискорити підготовку нейронної мережі.

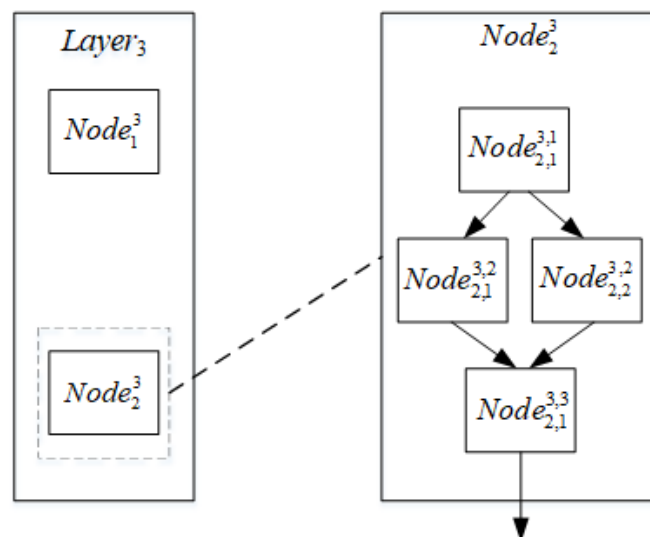


Рисунок 2.8 Структура $Node_2^3$

2.3.2 Основні структурні операції в мережі

Основні структурні операції в мережі перераховані нижче:

- Syn_{ADD} – додавання синапсу між двома випадково вибраними незв'язаними вузлами або мережами нейронів;
- Syn_{DEL} – видалення синапсу між двома випадковими незв'язаними вузлами або мережами нейронів;
- операція Syn_{MOVE} – переміщення синапсу між двома випадковим чином виділеними незалежними вузлами або мережами нейронів;
- операція AF_{MOD} – зміна функції активації нейрона на випадково відібраний нейрон;
- операції $Node_{SER}$ – серіалізація вузла;
- операції $Node_{PAR}$ – розпаралелювання вузла;
- операція $Node_{ADD}$ – додавання вузла;
- операція $Layer_{ADD}$ – створення нового шару;
- робота $Layer_{DEL}$ – видалення шару нейронної мережі.

Розглянемо операції більш детально. Мережа встановлена як такий граф:

$$G = \{\overline{Node}, \overline{Syn}\}, \quad (2.11)$$

де \overline{Node} - сукупність вузлів, що формують множину слоїв \overline{Layer} , а \overline{Syn} – набір синапсів, які їх зв'язують вузли між собою, при чому кожен синапс з'єднує дві ноди графу та являє собою пару нод:

$$(\forall Syn_{AB} \in \overline{Syn}) \exists \langle Node_A, Node_B \rangle, \quad (2.12)$$

де $Node_A \in \overline{Node}$ та $Node_B \in \overline{Node}$.

Операція Syn_{ADD} – додавання синапсу між двома випадковим чином виділеними незалежними вузлами показана на рисунку 2.8.

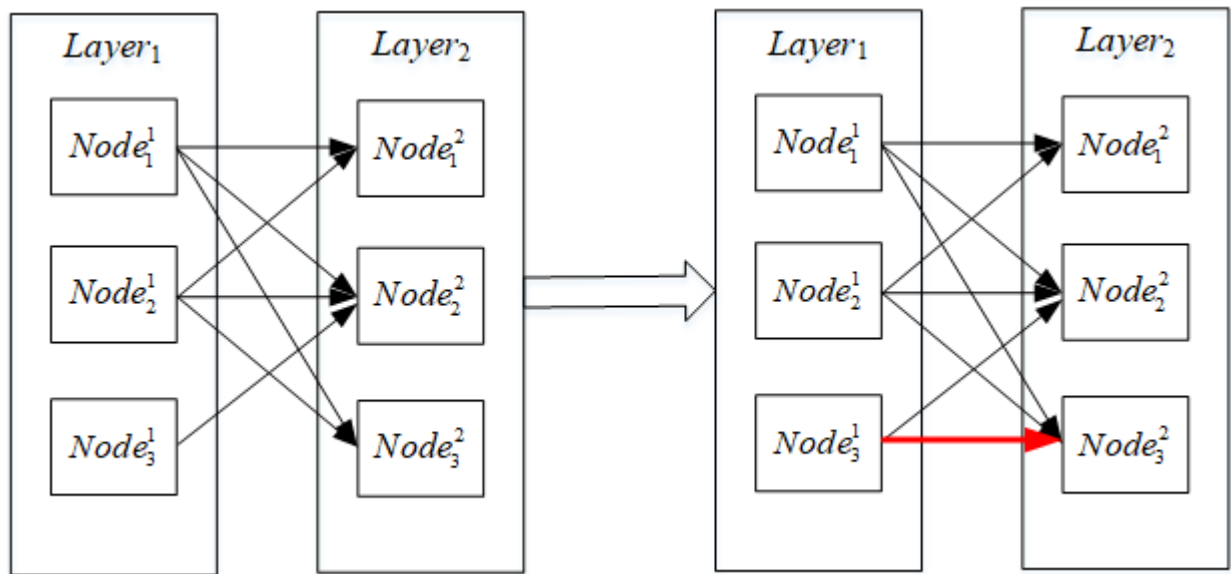


Рисунок 2.9 Приклад операції $Syn_{ADD}(Node_3^1; Node_3^2)$ – додавання синапсу

Операція Syn_{ADD} створює новий синапс між вже існуючими нодами. При чому, синапс створюється лише між сусідніми шарами або в межах однієї складної ноди. Тоді:

$$\begin{aligned} Syn_{ADD} : \overline{Syn} &\rightarrow \overline{Syn}, \\ Syn_{ADD}(\overline{Syn}, A, B) &= \overline{Syn} \cup \langle Node_A, Node_B \rangle, \end{aligned} \quad (2.13)$$

де $Node_A \in \overline{Node}$ та $Node_B \in \overline{Node}$.

Операція видалення синапсу між двома випадково виділеними вузлами або мережами, пов'язаними з нейронами, показана на рисунку 2.9.

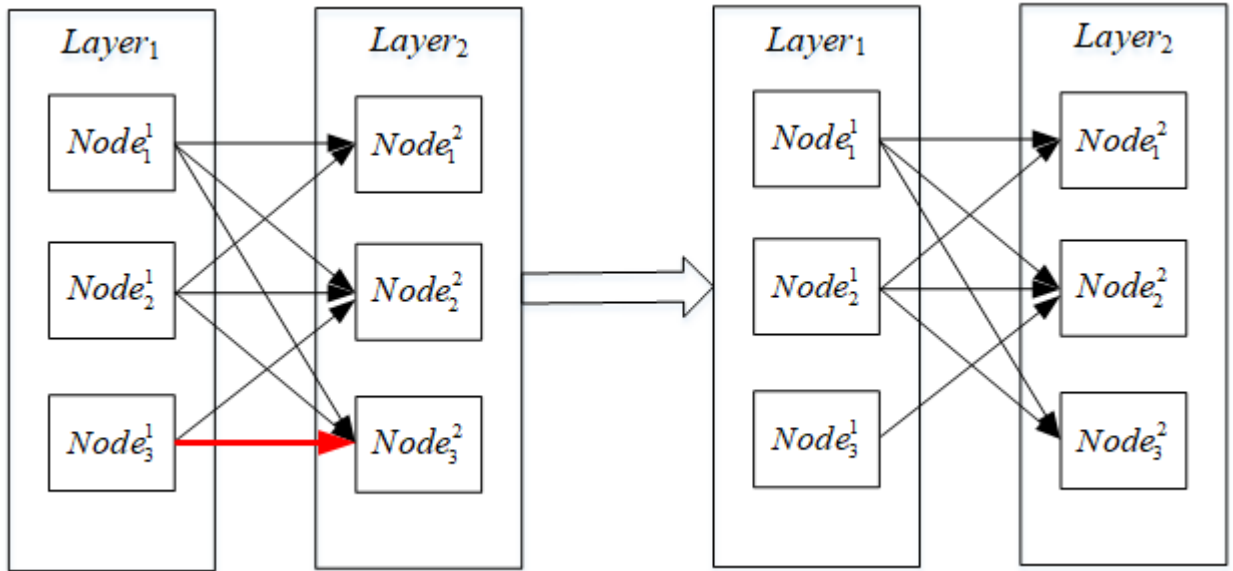


Рисунок 2.10 Приклад операції $Syn_{DEL}(Node_3^1; Node_3^2)$ –видалення синапсу

Операція Syn_{DEL} видаляє синапс між вже існуючими нодами. При чому, синапс видаляється лише між сусідніми шарами або в межах однієї складної ноди. Тоді:

$$\begin{aligned}
 Syn_{DEL} : \overline{Syn} &\rightarrow \overline{Syn}, \\
 Syn_{DEL}(\overline{Syn}, A, B) &= \overline{Syn} \setminus \langle Node_A, Node_B \rangle,
 \end{aligned}
 \tag{2.14}$$

Дія руху синапсу між двома випадковим чином виділеними вузлами або нейронною мережею показана на рисунку 2.10.

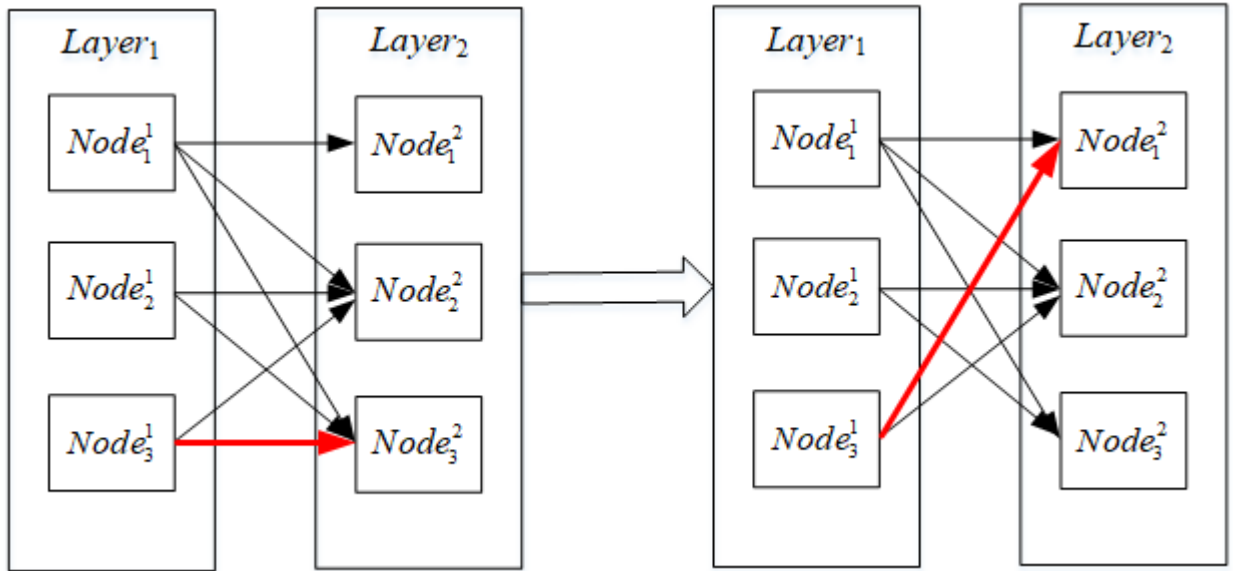


Рисунок 2.11 Приклад операції переміщення синапсу

За такої операції другий елемент пари в синапсі змінюється на іншу ноду, тобто:

$$\begin{aligned}
 Syn_{MOVE} : \overline{Syn} &\rightarrow \overline{Syn}, \\
 Syn_{MOVE}(\overline{Syn}, A, B, C) &= \overline{Syn} \setminus \langle Node_A, Node_B \rangle \cup \langle Node_A, Node_C \rangle,
 \end{aligned} \tag{2.15}$$

Результат $Node_{SER}$ випадково вибраного вузла або нейрона мережі показано на малюнку 2.11

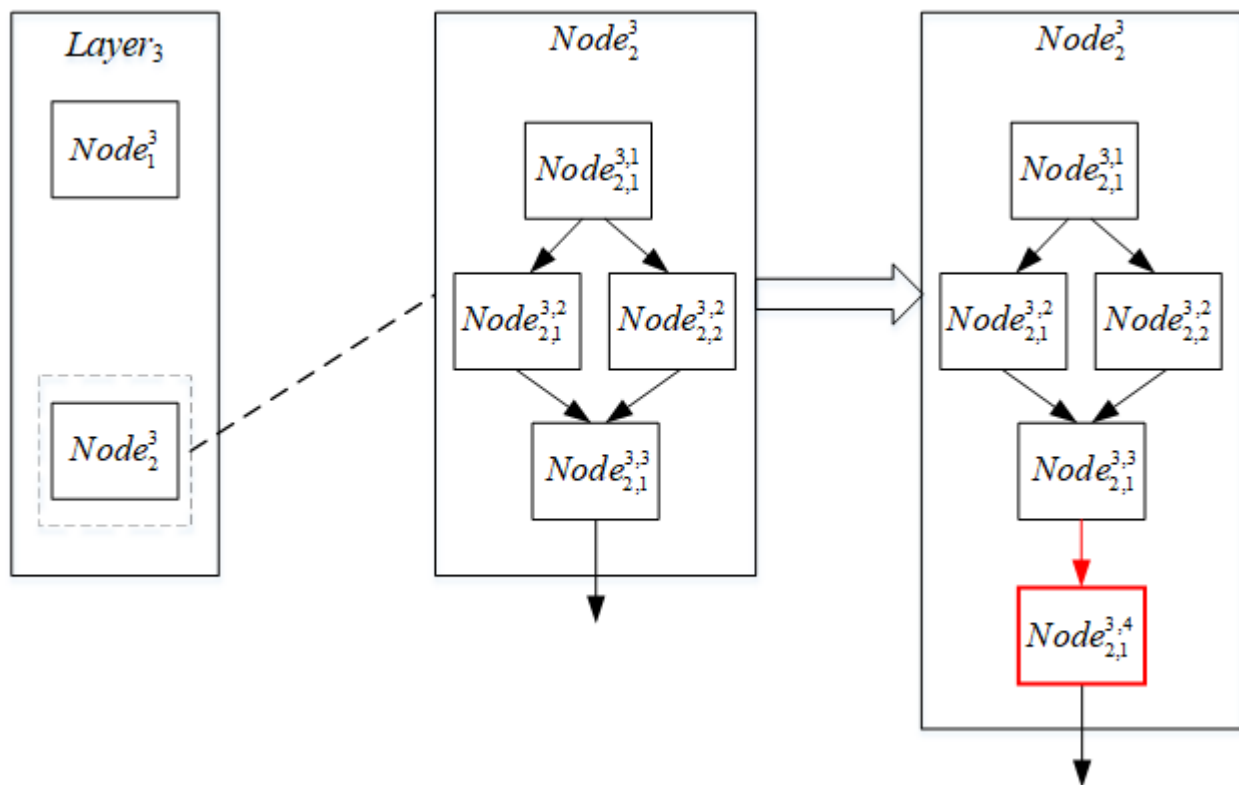


Рисунок 2.12 Операція створення послідовного нейрону

За такої операції додається новий нейрон послідовно вже існуючому. При чому, другий елемент пари в синапсі змінюється на іншу ноду, тобто:

$$\begin{aligned}
 &Node_{SER} : \overline{Syn} \rightarrow \overline{Syn}, \\
 &Node_{SER}(\overline{Syn}, A, B, C) = \overline{Syn} \setminus \langle Node_A, Node_B \rangle \cup \\
 &\cup \langle Node_A, Node_C \rangle \cup \langle Node_C, Node_B \rangle,
 \end{aligned} \tag{2.16}$$

де $Node_C$ – нова нода, яку було додано послідовно ноді $Node_B$.

Робота паралелізації $Node_{PAR}$ випадково вибраного вузла показана на малюнку 2.12.

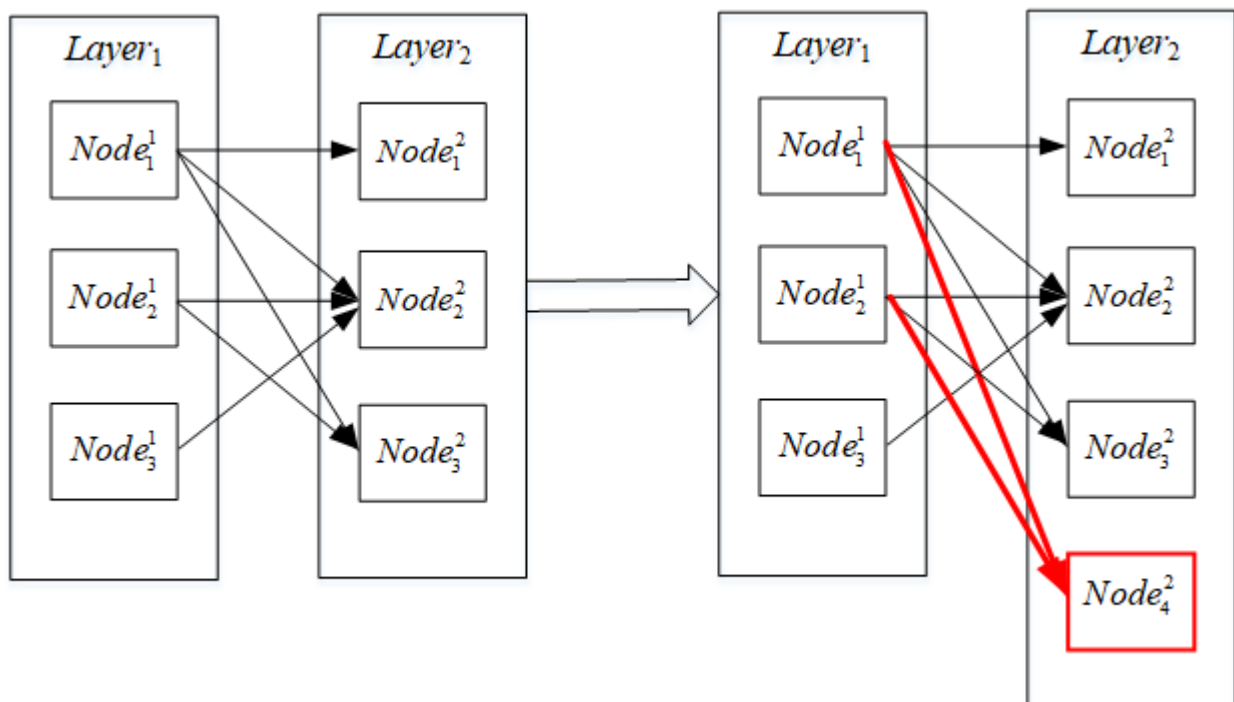


Рисунок 2.13 Операція створення паралельного нейрону

За такої операції додається новий нейрон паралельно вже існуючому. При чому, всі синапси, що входили в вузол, що копіюється, дублюються для нового вузла:

$$\begin{aligned}
 &Node_{PAR} : \overline{Syn} \rightarrow \overline{Syn}, \\
 &(\forall C : Syn_{CA} \in \overline{Syn}) \exists Syn_{CB} : Node_{PAR}(\overline{Syn}, A, B) = \\
 &= \overline{Syn} \cup \langle Node_C, Node_B \rangle,
 \end{aligned}
 \tag{2.16}$$

де $Node_C$ – вузли з яких виходили синапси до вузла $Node_A$, $Node_B$ – новий вузол-копія вузла $Node_A$, що паралельний вузлу $Node_A$.

Операція додавання вузла або нейрона $Node_{ADD}$ показана на малюнку 2.13.

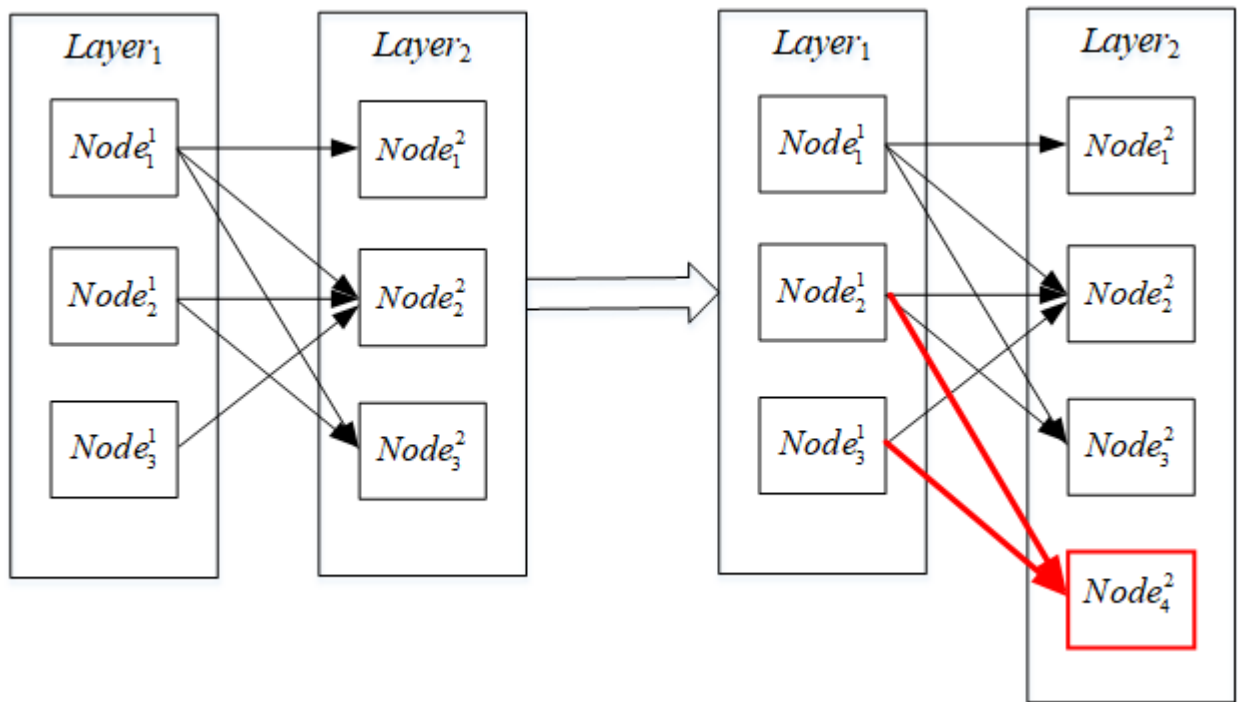


Рисунок 2.14 Операція додавання нейрону зі створенням нових зв'язків

При додаванні нового вузла, до та від нового вузла створюються випадкові синапси. Видалення вузла виконується з одночасним видаленням всіх синапсів, спряжених з вузлом. Всі синапси, що виходять з цього елемента, випадковим сполученням з елементами, а потім видаляються. Операція створення нового шару є модифікатором для операцій з додавання вузла або нейрона та операцій серіалізації та розпаралелювання, описаних вище, і вказує на те, що результат операції додає новий шар мережі.

Для прикладу розглянуто алгоритм структурної оптимізації в навчанні. Як основний алгоритм навчання, взято алгоритм зворотного поширення помилки, запропонований в [2]. Нейронна мережа має сигмоїдну функцію активації:

$$f(u) = \frac{1}{1 + e^{-u}}, \quad (2.17)$$

де

$$u = w_0 + w_1x_1 + \dots + w_nx_n. \quad (2.18)$$

Позначимо сукупність значень вихідних нейронів нейронної мережі через $F_i = \{f_{ij}\}$, де i – номер шару а j – кількість нейронів у шарі. Для вхідного шару $j = \overline{1, P}$, для першого прихованого шару – $j = \overline{1, C_1}$, другого - $j = \overline{1, C_2}$ і так далі, для останнього прихованого шару $j = \overline{1, 3}$. Загальна кількість шарів нейронної мережі дорівнює K . Алгоритм зворотного поширення помилки може бути представлений наступним чином:

Алгоритм зворотного поширення помилки може бути представлений наступним чином:

1. Для вхідного шару встановлюється значення кожного елемента відповідно до вхідного вектору. Значення виходу кожного елемента, що встановлено, дорівнює входу.

$$F_0 = X_1, \quad (2.19)$$

2. Для першого шару нейронів обчислюється загальний вхід та вихід

$$u_j^1 = w_{j0}^1 + \sum_{i=1}^N f_i^0 w_{ji}^1, \quad (2.20)$$

де

$$f_j^1 = \frac{1}{1 + e^{-u_j^1}}, \quad (2.21)$$

де $j = 1, 2, \dots, N_1$ – номери нейронів першого прихованого шару, w_{ij}^l – i -й ваговий коефіцієнт j -го нейрона l -шару нейронної мережі, f_i^0 – значення i -го нейрону у вхідному шарі.

3. Перший крок повторюється для всіх прихованих шарів нейронної мережі, включаючи вихідний шар нейронів, але формули (2.20) та (2.21) видозмінюються у:

$$u_j^k = w_{j0}^k + \sum_{i=1}^{U_k} f_i^{c-1} w_{ji}^c, \quad (2.22)$$

де

$$f_j^c = \frac{1}{1 + e^{-u_j^c}}, \quad (2.23)$$

де $c=2, \dots, K$.

4. Значення векторів Y_0 і F_K порівнюються, якщо різниця між векторною моделлю Y_0 і реальним виходом нейронної мережі F_K знаходяться у прийнятному діапазоні, тоді виконується крок 5, якщо ні, то виконується крок 6.

5. На вхід нейронної мережі подається наступне значення з навчальної вибірки і виконується крок 2. Якщо на вхід подається останній елемент навчальної вибірки, то на вхід нейромережі подається перший елемент навчальної вибірки.

6. Для кожного нейрону з вихідного шару обчислюється помилка. Оскільки нейронна мережа використовує сигмоїдну функцію активації, помилка дорівнює:

$$\delta_i = \begin{cases} f_i^1(1-f_i^1)(y_i - f_i^1), & \text{при } f_i < f_i^1 \\ f_i(1-f_i)(y_i - f_i), & \text{при } f_i^2 < f_i < f_i^1, \\ f_i^2(1-f_i^2)(y_i - f_i^2), & \text{при } f_i > f_i^2 \end{cases} \quad (2.23)$$

де f_i - результуюче значення вихідного шару j -го елемента; f_i^2, f_i^1 - локальна крайня точка до очікуваного значення y_i в діапазоні $[0,1]$.

7. Обчислюється коефіцієнт мережових мутацій:

$$M_{NET} = \frac{1}{\delta(t+1)}, \quad (2.24)$$

Залежно від коефіцієнтів, модифікація структури нейронної мережі здійснюється за допомогою основних структурних операцій, обговорюваних раніше (табл. 2.1). Тенденція оціночного коефіцієнту вказує на динаміку навчання. Якщо в кожному навчальній епоху вона зростає, це означає, що ця помилка зменшується, немає сенсу щось змінювати у структурі мережі. І навпаки, якщо тенденція падає, вам потрібно змінити структуру мережі.

Таблиця 2.1 Використання базових структурних операцій

Значення M_{NET}	Тенденція M_{NET}	Операцію, яку треба виконати
>1	Зростає	Нічого не робити
>1	Спадає	L_{ADD}
>100	Зростає	Нічого не робити
>100	Спадає	$Node_{SER}, Node_{PAR}, Node_{ADD}$
>500	Зростає	Нічого не робити
>500	Спадає	$Syn_{ADD}, Syn_{MOVE}, AF_{MOD}$

8. Для підрахунку передостаннього шару помилка кожного нейрона обчислюється за формулою:

$$\delta_1^{K-1} = f_i^{K-1} (1 - f_i^{K-1}) \sum_{j=1}^{c_K} \delta_j^K w_{ij}^{K-1}, \quad (2.25)$$

де δ_j^K – похибка j -го елемента K -го шару, w_{ij}^{K-1} – вага зв'язку між i -м і j -го елементами $K-1$ -го шару.

9. Для всіх інших прихованих шарів помилок обчислюється за формулою (16).

10. Для всіх шарів вагові коефіцієнти кожного нейрона оновлюються. елемента K -ий шар, f_i^{K-1} – значення i -го елемента $K-1$ -го шару.

$$\Delta w_{ij}^c(t+1) = \eta(\delta_i^c f_j^{c-1}) + \alpha \Delta w_{ij}^c(t), \quad (2.26)$$

де η - швидкість тренування, α - інерція, або вплив попередніх змін, як правило, $\alpha < 1$, t – кількість ітерацій. Потім встановлюється нове значення коефіцієнтів ваги, рівне:

$$w_{ij}^c = w_{ij}^c + \Delta w_{ij}^c(t). \quad (2.27)$$

11. Для вхідних векторів встановлюється значення кожного нейрона відповідно до елементів навчальної вибірки і алгоритм переходить до кроку 2. Якщо останній векторі навчальної вибірки вже подано на вхід нейронної мережі, то повторно оброблюється перший.

2.4 Метод нечіткої непараметричної оцінки якості сервісу

При оцінці роботи кожного сервісу множини $S = \{S_o \mid o = \overline{1, \dots, O}\}$ необхідно враховувати значення параметрів $P' = \bigcup_{\eta, \varpi_\eta, o} P'_{\eta \varpi_\eta o}$, що впливають на якість ІТ-послуг, що входять до складу ІТ-сервісів. Протягом заданого інтервалу часу T виробляється M вимірів значень параметрів множини $P = \bigcup_{\eta, \varpi_\eta, o} P_{\eta \varpi_\eta o}$. Всі значення нормуються і наводяться до відрізка $[0, 1]$.

2.4.1 Обґрунтування концептуальних положень запропонованого методу.

У зв'язку з відсутністю припущень про характер залежностей Θ (2.6), а також можливості існування будь-якого роду відносин між ІТ-послугами сервісу, доцільно застосовувати наближені експертні оцінки, засновані на особистому досвіді адміністраторів, керівників ІТ-підрозділів і ін. Оцінки такого роду залежать від функціонального навантаження ІКТ, визначаються при

розгортанні ІТ-інфраструктури і її налаштування і задаються за допомогою функцій належності апарату нечіткої логіки [27]. Для аналізу часових залежностей між параметрами окремих елементів використовуються геометричні оцінки апарату непараметричної статистики, проекція якої на зони, що визначають якість функціонування, породжує узагальнену оцінку параметрів елементів. При цьому за допомогою апарату нечіткого виведення з урахуванням впливу якості складових ІТ-послуг обчислюється рівень якості ІТ-сервісу. Пропонований двоетапний метод дозволяє отримати статистичну оцінку якості роботи сервісу без будь-яких припущень про імовірнісні характеристики процесів, що протікають в ІТ-інфраструктурі, а також узагальнити її на основі досвіду експертів, спроектувавши на якісну вісь. Подання елементів у параметричному просторі станів вимагає аналітичної обробки даних в евклідовому просторі, яка дозволяє виділити зону найбільш характерного функціонування, відповідну центральному регіону. Із визначень [28] центральних упорядкованих регіонів найбільш придатною є нотація зоноїда [29] дякуючи ряду корисних властивостей, серед яких слід виділити афінну еквіваріантність, "прив'язує" оцінку положення до елементів, повноту інформації, що забезпечує унікальність оцінки, безперервність по глибині - параметру, який визначає центральність регіону, і розподілу, що забезпечує стійкість рішення до вхідних даних, опуклість, яка спрощує розрахунок часток приналежності, а також наявність алгоритмічного апарату побудови самого зоноїда, що є опуклим багатогранником.

Оскільки в роботі розглядаються гомогенні сервери, а віртуальні машини $V_\eta = \{V_{\eta\varpi_\eta o} \mid \varpi_\eta = \overline{1, \dots, \zeta_\eta}, o \in [1, O]\}$, що надають сервіс мають однакові характеристики, то при визначенні залежності Θ (2.6) індекс o можна тимчасово виключити з розрахунків.

2.4.2 Розрахунок рівня якості ІТ-сервісу

Припустімо, на рівень якості сервісу $S \in S$ впливають параметри $P \in P$. Діапазони значень рівня якості сервісів розбиваються на піддіапазони, що відповідають класам A , B та C . Оцінка Q сервісу і параметри якості $p_1, p_2, \dots, p_{|P|}$ сервісу описуються лінгвістичними значеннями з множин $\{l_{i,\omega}^Q\}$, $i = \overline{1, K}$, $\omega = \{A, B, C\}$ и $\{l_{b,i,\gamma}^q\}$, $\gamma = \{A, B, C\}$. Кожному $l_{i,\omega}^Q$, $\omega = \{A, B, C\}$ и $\{l_{b,i,\gamma}^q\}$, $\gamma = \{A, B, C\}$ ставляться у відповідність нечіткі множини $\Psi_{i,\omega}^Q$ и $\Psi_{b,i,\gamma}^q$.

Визначимо рівень якості окремої ІТ-послуги. Для цього протягом інтервалу T значення параметрів з множини P заміряються M раз, в результаті чого формується M множин такого вигляду

$$P_{a,m} = \{p_{a,1,m}, p_{a,2,m}, \dots, p_{a,D_a,m}\}, \text{ для всіх } m = \overline{1, M} \quad (2.28)$$

Розглянемо L_i -вимірний простір L_i , в якому кожна вісь відповідає окремому параметру, а кожному заміряному значенню параметру відповідає точка простору, що відображає стан параметрів під час заміру значень параметрів. Для емпіричного розподілу на просторі L_i зоноїд визначається як [11]

$$Z(\alpha) = \text{conv}\{U_1, U_2, \dots, U_M\}, \quad (2.29)$$

для $\alpha \in]0, \frac{B}{M}]$ та

$$\text{conv}\left\{\frac{1}{\alpha \cdot M} \sum_{\beta=1}^B U_{p_\beta} + \left(1 - \frac{1}{\alpha \cdot M}\right) U_{p_{B+1}}\right\}, \quad (2.30)$$

для $\alpha \in [\frac{B}{P}, \frac{B+1}{P}]$, де $B \in \{1, 2, \dots, M-1\}$, $\{p_1, p_2, \dots, p_{B+1}\} \subset \{1, 2, \dots, M\}$, а $\{U_1, U_2, \dots, U_M\}$ – точки, на базі яких будується зоноїд, M – кількість таких точок, α – глибина зоноїда.

На етапі фазифікації визначається ступінь належності $L_{b,i,\gamma}^q$ значення показника якості $q_{b,i}, \forall b, i$ віртуальної машини до нечітких множин $\Psi_{b,i,\gamma}^q$. Для сервісу визначається L_i -вимірний простір, де кожній осі ставиться у відповідність окремий параметр множини $P = \bigcup_{\eta, \varpi_\eta, o} P_{\eta \varpi_\eta o}$. Отриманий регіон виділяє R_{L_i} область, в якій, зі статистичної точки зору, найбільша вірогідність перебування точки, що характеризують стан роботи ІТ-послуги, а глибина регіону пропорційна ризику виходу таких точок за межі $Z_a(\alpha)$. Так як всі осі L_i -вимірного простору, що розглядається, розбиті на діапазони, то гіперплощини, що перпендикулярні координатним осям і проходять через точки кордонів таких діапазонів, розбивають отриманий зоноїд на опуклі замкнуті L_i -вимірні частки. Кожній частці відповідає певне висловлювання з нечіткою бази знань, на підставі яких і визначається рівень якості ІТ-послуги.

Для всіх $b = 1, B_a$ розраховується сума відносних L_i -мірних об'ємів долей зоноїда, що належать підпросторам, що відповідають заданим нечітким висловлюванням:

$$V_{a,b} = \frac{\sum_{j=1}^{R_{a,b}} V_{a,b,j}}{V_a}, \quad (2.31)$$

де $R_{a,b}$ – кількість неповторюваних нечітких висловлювань виду $\bigcap l_{b,i,\gamma}^q$ правил бази знань, вихідний величиною яких є лінгвістичних значень, V_{abj} L_i – вимірний об'єм випуклої замкненої долі зоноїда, що відповідає висловлюванню з бази правил, а V_a – L_i -вимірний об'єм $Z_a(\alpha)$.

Q , визначається центроїдним методом:

$$Q_i = \frac{\int_{x=0}^1 x \cdot F_i(x) dx}{\int_{x=0}^1 F_i(x) dx}, \quad (2.32)$$

де $F_i(x)$ розраховується за формулою:

$$F_i(x) = \text{agg} \left(\text{imp} \left(L_{i,\omega}^Q, \mu_{i,\omega}(x) \right) \right), \quad (2.33)$$

де $\mu_{i,\omega}$ – функція належності показника якості від до нечіткої множини $\Psi_{i,\omega}^Q$.

3.1 Розроблення узагальненої схеми управління ІТ-інфраструктурою

Узагальнена схема управління ІТ-інфраструктурою для підтримки якості ІТ-послуг на узгодженому рівні наведена на рисунку 3.1. У контурі управління ІТ-інфраструктурою виділено два модулі – модуль оцінки якості та модуль управління.

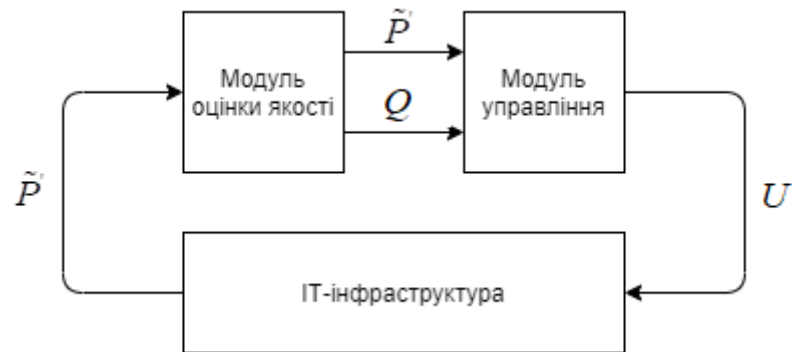


Рисунок 3.1 Узагальнена схема управління ІТ-інфраструктурою

Для підтримання якості Q^* надання ІТ-послуг на узгодженому рівні при задіянні мінімальної кількості ресурсів \mathfrak{R} модуль контур управління ІТ-інфраструктурою, реалізуючи запропонований у роботі метод управління рівнем якості ІТ-послуг, генерує множину керуючих впливів U на ресурси ІТ-інфраструктури \mathfrak{R} . Визначення керуючих впливів здійснюється виходячи з аналізу значень параметрів P' , розрахованих оцінок якості Q та значень узгоджених рівнів якості послуг Q^* . Управління здійснюється таким чином, щоб виконувалась умова:

$$|Q_{eo\mathfrak{k}} - Q_{eo\mathfrak{k}}^*| \rightarrow 0 \forall e = \overline{1..E}, o = \overline{1, \dots, O}, \mathfrak{k} \in \mathfrak{K}_{eo}. \quad (3.1)$$

Після того як визначена поточна якість послуг $Q = \{Q_{eot} \mid e = \overline{1..E}, o = \overline{1, ..., O}, t \in \mathcal{K}_{eo}\}$, отримані оцінки, а також множина значень параметрів \tilde{P} надсилаються на модуль управління для генерації множини керуючих впливів U (рис 3.1). Можливі керуючі впливи належать множині усіх керуючих впливів U :

$$U = \{U_{eop} \mid p \in [1, ..., P_e], e \in [1..E], o \in [1, ..., O]\}, \quad (3.2)$$

де U_{eop} – керуючий вплив на зміну обсягів віртуальної машини для зміни значення p -го параметру, що пов'язаний з показником якості e -ї послуги, яка забезпечує надання o -го сервісу.

Керування може відбуватися в умовах надлишку або дефіциту ресурсів \mathcal{R} ІТ-інфраструктури. У першому випадку ресурсів \mathcal{R} ІТ-інфраструктури достатньо для забезпечення належного рівня якості всіх послуг. Тому керування ІТ-інфраструктурою здійснюється таким чином, щоб для підтримання узгодженого рівня якості усіх послуг використовувалась мінімальна кількість ресурсів. У другому випадку, модуль керування має перерозподілити ресурси \mathcal{R} таким чином, щоб більш пріоритетні послуги функціонували з забезпеченням узгодженого рівня, а надання менш пріоритетних послуг відбувається з фактичним рівнем якості або припиняється в загалі.

3.2 Управління ресурсами ІТ-інфраструктури, що надаються для функціонування ІТ-послуги

В даному розділі розглядається задача: необхідно генерувати керуючі впливи на ресурси \mathcal{R} ІТ-інфраструктури таким чином, щоб виконувалася умова (3.1) для послуги $\hat{e} \in E$.

Нехай вимоги до рівня якості послуги формують множину $Q_{\hat{e}}^* = \{Q_{\hat{e}t}^* \mid t \in \mathcal{K}_{\hat{e}}\}$. Модуль оцінки якості розраховував поточний рівень якості даної

послуги $Q_{\hat{e}} = \{Q_{\hat{e}\mathfrak{k}} \mid \mathfrak{k} \in \mathfrak{K}_{\hat{e}}\}$ за всіма узгодженими показниками якості обраного сервісу, таким чином, залежність $q_{\hat{e}}$ (2.9) вже відома.

Для цього у $|P_{\hat{e}}|$ -мірному просторі параметрів послуги \hat{e} виділяються зона, яка обмежена значеннями параметрів, керувати якими СУІ не має змоги. Дана зона обмежується значеннями параметрів, що відповідають вимогам до якості $Q_{\hat{e}}^* = \{Q_{\hat{e}\mathfrak{k}}^* \mid \mathfrak{k} \in \mathfrak{K}_{\hat{e}}\}$. Тоді, вирішення поставленої задачі зводиться до знаходження точки у цьому просторі, яка одночасно належить даній зоні простору та гіперповерхні, що формується $q_{\hat{e}}$. Будь-яка точка на цій гіперповерхні задовільняє умові (3.1), а знаючи координати таких точок можна легко визначити які та скільки ресурсів треба додатково задіяти, або вивільнити для досягнення послугою \hat{e} узгодженого рівня якості за всіма критеріями з множини $\mathfrak{K}_{\hat{e}}$.

Для наглядності на рис. 3.2 наведено приклад залежності якості послуги $Q_{\hat{e}}$ від значень параметру $p_{\hat{e}}$. При чому якість послуги залежить лише від даного параметру і СУІ може керувати ІТ-інфраструктурою таким чином, що значення цього параметру можуть змінюватися. Нехай поточне значення параметру рівне $p_{\hat{e}}^*$. При такому значенні рівень якості послуги є незадовільним. Але при значенні параметру $p_{\hat{e}}$, що потрапляє у діапазон $[p_{\hat{e}}', p_{\hat{e}}'']$, рівень якості послуги буде на узгодженому рівні.

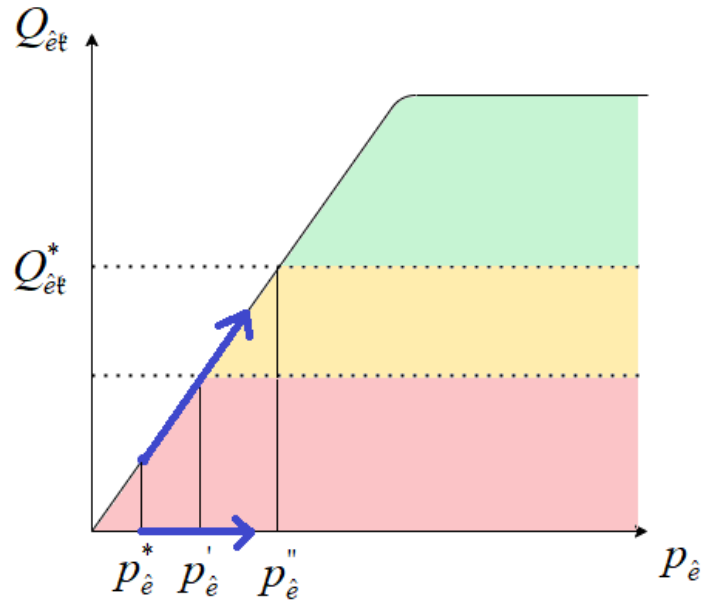


Рисунок 3.2 Залежність якості послуги \hat{e} від параметру $p_{\hat{e}}$

Моделі, що було побудовано у розділі 2 дають можливість визначати лише залежність $q_{\hat{e}}$, але не дають можливості вирішити зворотню задачу, тобто, знайти залежність $p_{\hat{e}}$.

Для знаходження $p_{\hat{e}}$ необхідно визначити множини фіксованих точок у гіперпросторі параметрів $E_{\hat{e}}$:

$$\underline{P}_{\hat{e}} = \left\{ \underline{P}_{\hat{e}\rho v} \mid \rho = \overline{1 \dots P_{\hat{e}}}, v = \overline{1 \dots V_{\hat{e}}} \right\}, \quad (3.3)$$

де $V_{\hat{e}}$ – кількість точок, що було обрано для послуги $E_{\hat{e}}$.

Розраховується значення:

$$\underline{Q}_{\hat{e}}^* = \left\{ q_{\hat{e}} \left(\underline{P}_{\hat{e}\rho v} \right) \mid \rho = \overline{1 \dots P_{\hat{e}}}, v = \overline{1 \dots V_{\hat{e}}} \right\}. \quad (3.4)$$

Далі необхідно знайти точку множини, при якій виконується умова

$$\left| \underline{Q}_{\hat{e}} - Q_{\hat{e}}^* \right| \rightarrow \min. \quad (3.5)$$

Для кожного елементу множини впливів $\{U_{\hat{e}\rho} \mid \rho \in [1, \dots, P_{\hat{e}}]\}$ вираховується параметр виконавчого скрипту як різниця між поточним значенням параметру та значенням параметру у точці множини $\underline{P_{\hat{e}}}$, що задовольняє умові (3.5).

3.3 Управління ресурсами при балансуванні навантаженням на елементи ІТ-інфраструктури

В даному розділі вирішується задача: при використанні балансувальника навантаження, що розподіляє запити користувачів між множиною гомогенних віртуальних машин $V_{\hat{e}}$ для надання послуги $\hat{e} \in E$, визначити чи є необхідність створювати додатковий, або видаляти існуючий екземпляр віртуальної машини що надає послугу \hat{e} .

Питання про планування навантаження слід вирішувати ще на стадії проектування проекту. При помірному збільшенні навантаження на сервери ІТ-інфраструктури проблему недостатньої продуктивності, що виникає при зростанні навантаження, можна вирішити шляхом використання більш продуктивних серверів або покращення використовуваних алгоритмів. Але в якийсь момент перед ІТ-підрозділом постане задача розподілення навантаження між декількома серверами. Навантаження між ними розподіляється за допомогою комплексу спеціальних методів, які називаються балансуванням. Крім вирішення проблеми високих навантажень такий підхід також може забезпечити резервування серверів один на одного.

Ефективність розподілення навантаження безпосередньо залежить від того, як розподіляється (балансується) навантаження між елементами ІТ-інфраструктури. Балансування навантаження може здійснюватися за допомогою як апаратних, так і програмних інструментів.

В роботі розглядається випадок коли балансувальник навантаження розподіляє запити користувачів між екземплярами гомогенних віртуальних

машин, а кожна віртуальна машина виконує заздалегідь заплановані функції, що описані у інтерфейсі прикладного програмування (API).

Існує багато алгоритмів роботи балансувальника: розподіл запитів користувачів послідовно по віртуальним машинам, направлення нового запиту користувача на віртуальну машину у якої мінімальна черга та ін. Конкретний Алгоритм роботи балансувальника визначається ІТ-підрозділом. Будемо вважати що алгоритм обрано відповідно до специфіки послуги \hat{e} , а якість надаваних послуг відповідає узгодженому рівню при умові що для обробки запитів користувача ресурсів віртуальних машин множини $V_{\hat{e}}$ достатньо.

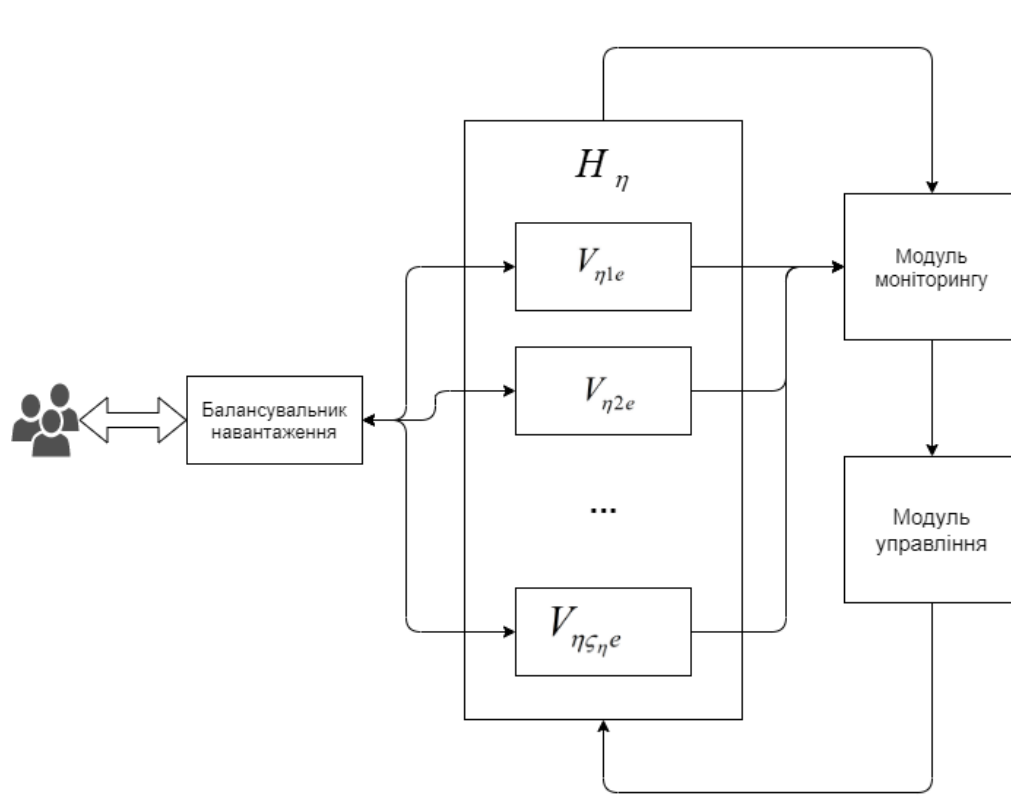


Рисунок 3.3 Використання балансувальника навантаження у системі управління ІТ-інфраструктурою

За таких умов СУІ має вчасно виконувати дві задачі: перша – визначити момент часу, коли виділених ресурсів не достатньо для надання послуги на узгодженому рівні якості і необхідно створити новий екземпляр віртуальної машини, а друга – визначити коли є можливість видалити екземпляр однієї з

віртуальних машин для того щоб вивільнити додаткові ресурси ІТ-інфраструктури.

Позначимо як $\mathcal{Z}_{\hat{e}}$ множину запитів користувачів на віртуальні сервери з множини $V_{\hat{e}}$, а $\mathcal{R}_{\hat{e}}$ – ресурси ІТ-інфраструктури, що задіяні для підтримання роботи $V_{\hat{e}}$, а $\hat{t}_{V_{\hat{e}}}$ – час, за який можна створити новий екземпляр віртуальної машини з множини $V_{\hat{e}}$

$$\begin{aligned} V_{\hat{e}} &= \{V_{\hat{e}\hat{\omega}} \mid \hat{\omega} = \overline{1..\zeta_{\hat{e}}}\}, \\ \mathcal{Z}_{\hat{e}} &= \{\mathcal{Z}_{\hat{e}i_{\hat{e}}} \mid i_{\hat{e}} = \overline{1..I_{\hat{e}}}\}, \\ \mathcal{R}_{\hat{e}} &= \{\mathcal{R}_{\hat{e}\hat{\omega}} \mid \hat{\omega} = \overline{1..\zeta_{\hat{e}}}\}, \end{aligned} \quad (3.6)$$

де $\mathcal{R}_{\hat{e}\hat{\omega}}$ – ресурси ІТ-інфраструктури, що задіяні для роботи $\hat{\omega}$ -ї віртуальної машини $V_{\hat{e}\hat{\omega}}$, $\zeta_{\hat{e}}$ – кількість задіяних віртуальних машин для надання послуги \hat{e} , $I_{\hat{e}}$ – кількість користувачів, що користуються даною послугою.

Для вирішення поставленої задачі необхідно передбачити момент часу t_{new} , коли необхідно запустити процес створення нової віртуальної машини. Для цього необхідно визначити скільки в середньому запитів користувача та відкритих сеансів може обробити кожна машина множини $V_{\hat{e}}$ без втрати якості послуги, що вона надає. І передбачити в який момент часу необхідно запустити створення нової віртуальної машини з урахуванням часу її створення $\hat{t}_{V_{\hat{e}}}$.

Позначимо як $API_{\hat{e}}$ множину точок доступу, до яких має можливість звернутися користувач. Кожне звернення до точки доступу ініціює виконання певного, заздалегідь запрограмованого алгоритму, що прив'язаний до цієї точки доступу. Кожне звернення нового користувача створює сеанс між користувачем і віртуальною машиною, а кожна віртуальна машина має ліміт на кількість одночасно запусчених сеансів.

Для визначення поведінки середньостатистичного користувача зазвичай використовується один з двох алгоритмів. Перший, більш точний, можна

використовувати тільки при наявності історії запитів активних користувачів. Для цього аналізується історія звернень користувачів та визначається поведінка середньостатистичного користувача, тобто кількість та послідовність звернень до $API_{\hat{e}}$. Другий алгоритм менш точний, але його можна використовувати у випадку, якщо відсутня історія активних користувачів. Його суть полягає в тому, що для визначення поведінки середньостатистичного користувача генеруються випадкові запити на $API_{\hat{e}}$.

Враховуючи (2.11) дані два алгоритми знаходять залежність $p_{\hat{e}\hat{w}}$, за умови що залежність $p_{\hat{e}}$ відома:

$$p_{\hat{e}\hat{w}} : \mathcal{Z}_{\hat{e}\hat{w}} \rightarrow \mathfrak{R}_{\hat{e}\hat{w}} \forall \mathcal{Z}_{\hat{e}\hat{w}} \in \mathcal{Z}_{\hat{e}}, \quad (3.6)$$

тобто, яким чином користувач послугою \hat{e} впливає на віртуальну машину $V_{\hat{e}}$

Виходячи з (2.11) та (3.6),

$$p_{\hat{e}} : (\mathfrak{C}_{\hat{e}}(t), I_{\hat{e}}(t)) \rightarrow \zeta_{\hat{e}}(t), \quad (3.7)$$

де $\mathfrak{C}_{\hat{e}}(t)$ – якість послуги \hat{e} , у момент часу t , $I_{\hat{e}}(t)$ – кількість активних користувачів послугою \hat{e} на момент часу t , $\zeta_{\hat{e}}(t)$ – кількість віртуальних серверів, що обслуговують послугу \hat{e} на момент часу t .

Базуючись на визначеній поведінці $p_{\hat{e}\hat{w}}$ яку максимальну кількість користувачів може обробити віртуальна машина без зниження рівня якості. Це дасть змогу спрогнозувати час, коли необхідно почати створення нового екземпляру віртуальної машини щоб виконувалися умови:

$$\begin{cases} |Q_{\hat{e}}(t) - Q_{\hat{e}}^*(t)| \rightarrow 0, \\ |Q_{\hat{e}}(t + t_{new}) - Q_{\hat{e}}^*(t_{new})| \rightarrow 0. \end{cases} \quad (3.8)$$

Використання запропонованого підходу дозволить вчасно створювати додаткові віртуальні машини.

3.4 Керування якістю сервісу на основі апарату нечіткої логіки

Так як задача чітко розрахувати множину всіх впливів U на елементи ІТ-інфраструктури є дуже трудомісткою задачею, було вирішено оперувати наближеними нечіткими значеннями [41]. Такі значення є множиною лінгвістичних значень, що описують ресурс нечіткими характеристиками, які задає оператор, спираючись на особистий досвід. Тому розроблений метод управління ресурсами сервера дозволяє приймати рішення про те на скільки потрібно підвищити або понизити ресурси, керуючись апаратом нечіткої логіки (АНЛ). Схема АНЛ представлена на рисунку 2.6.

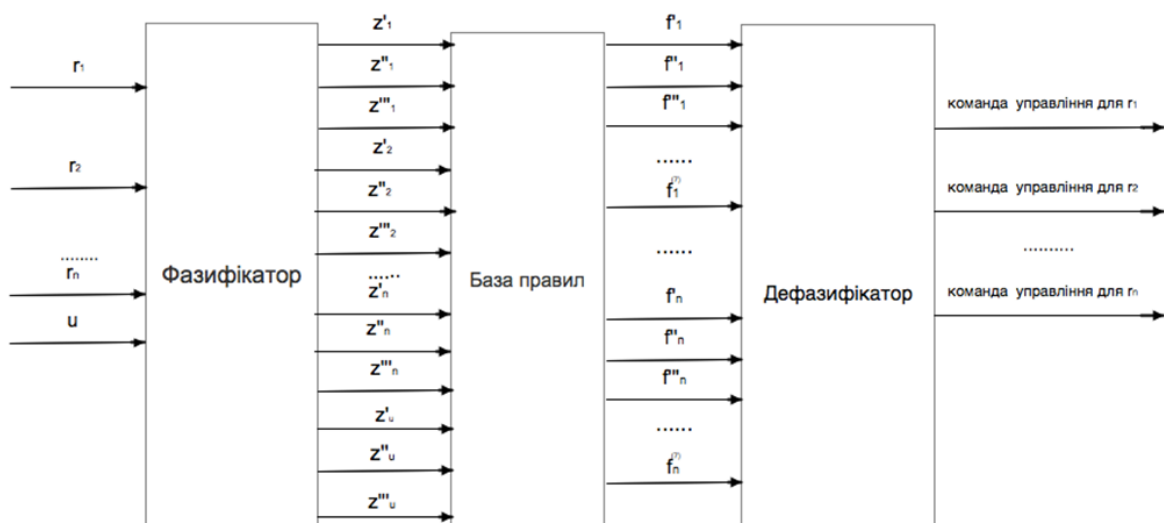


Рисунок 3.4 Схема АНЛ

АНЛ складається з таких частин: фазифікатор, база правил та дефазифікатор [42].

На вхід фазифікатору подається значення R_n ресурсу та кількість користувачів u , які в момент часу t користуються сервісом S_n . Цими ресурсами

може бути оперативна пам'ять сервера C_n , пропускна здатність каналу зв'язку сервера C_n або об'єм жорсткого диску сервера C_n . У фазифікаторі за допомогою функцій належності (2.10) визначається ступінь належності значення R_n до класу Z'_n, Z''_n або Z'''_n та кількості користувачів u до класу Z'_u, Z''_u або Z'''_u . Детальна схема фазифікатору для одного ресурсу R_l зображена на рисунку 2.7.

$$\forall Z_i \exists \mu_{Z_1} | \mu_{Z_1} : \mu_{Z_1}(R_n) = Z'_1, \quad (3.1)$$

де Z'_1 , – ступінь належності R_n до класу Z' , Z''_1 , – ступінь належності R_n до класу Z'' , Z'''_1 , – ступінь належності R_n до класу Z''' .

На вісь ОХ відкладається значення R_1 та через цю точку проводиться пряма l , яка паралельна осі ОУ. В точках, де пряма перетинає функції приналежності Z'_1, Z''_2 або Z'''_3 опускається перпендикуляр h_l на вісь ОУ. В точках, де перпендикуляр h_l перетинає вісь ОУ визначається ступінь приналежності значення кожного ресурсу R_n до класу Z'_1, Z''_1 або Z'''_1 .

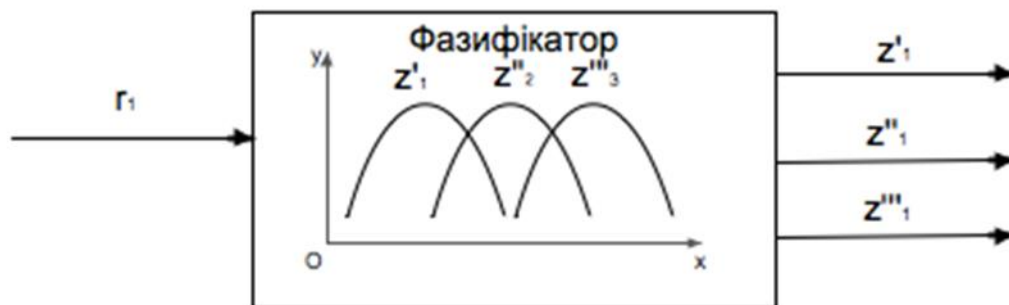


Рисунок 3.5 Фаза фазифікації ресурсів

Вектор Q , який визначає ступінь належності поточної якості w_n сервісу S_n до класу A, B або C вираховується за допомогою методу оцінки якості ІТ-послуг, що описаний у розділі 2.1.

База правил отримує на вхід значення Z'_n, Z''_n або Z'''_n та Z'_u, Z''_u або Z'''_u , які було отримано з фазифікатору та вектор Q . Схема представлена на рисунку 2.8.

На виході база правил дає числове визначення нечіткому визначенню *підвищити на багато, підвищити на середнє, підвищити на мало, нічого не робити, зменшити на багато, зменшити на середнє та зменшити на мало* [43]. Отже для кожного ресурсу $R_{\tilde{n}}$ визначається сім лінгвістичних змінних, які на схемі 2.8 представлені літерами $f'_n, f''_n, f'''_n, \dots, f_n^{(7)}$ відповідно.

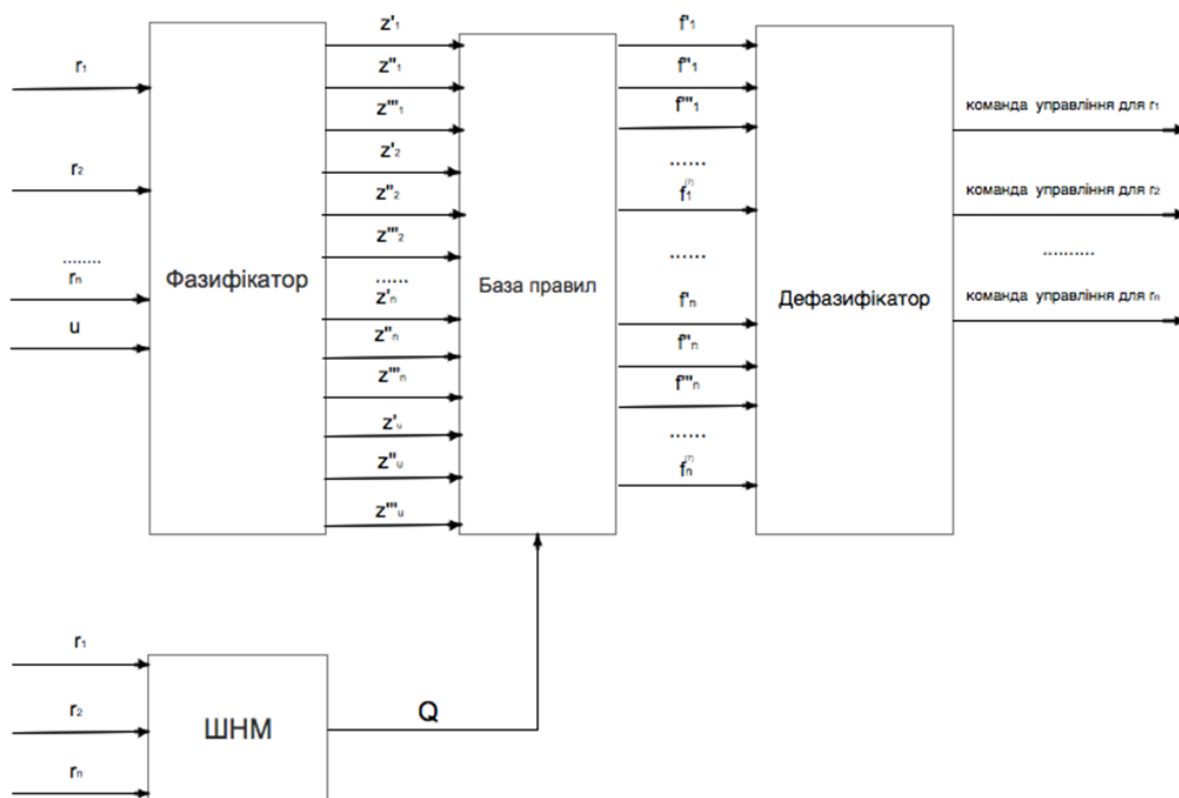


Рисунок 3.6 Схема ШНМ та апарату нечіткої логіки

В базі правил формується своя база правил для кожного ресурсу R_n . База правил виглядає наступним чином:

$$\left[\begin{array}{l} \text{Якщо } z'_1 \ z'_2 \ \dots \&z'_n q_1 \ z'_u, \text{ то понизити на багато} \\ \text{Якщо } z''_1 \ z'_2 \ \dots \ z'_n \ q_1 z'_u, \text{ то нічого не робити} \\ \text{Якщо } z''_1 \ z'_2 \ \dots \&z'''_n q_1 z'_u, \text{ то підвищити на багато} \end{array} \right. \quad (3.2)$$

Розглянемо на прикладі декілька правил за бази правил.

За ресурс R_1 обрано пропускну здатність каналу зв'язку (BAND), за ресурс R_2 обрано кількість використаного процесорного часу (CPU).

Приклади правил з бази правил для ресурсу пропускну здатність каналу зв'язку, процесорного часу, кількості користувачів та якості сервісу представлені на рисунку 2.9.

Якщо багато CPU & багато BAND & якість відмінна &
& кількість користувачів мало, то понизити ресурс BAND на багато.
Якщо багато CPU & середнє BAND & якість відмінна
& кількість користувачів мало, то понизити ресурс BAND на середнє.
Якщо багато CPU & мало BAND & якість відмінна
& кількість користувачів мало, то понизити ресурс BAND на мало.
Якщо багато CPU & багато BAND & якість задовільна &
кількість користувачів мало, то нічого не робити.
Якщо мало CPU & середнє BAND & якість відмінна
& кількість користувачів середнє, то понизити ресурс BAND на мало.
Якщо багато CPU & багато BAND & якість незадовільна &
& кількість користувачів багато, то нічого не робити

Рисунок 3.7 Приклад правил в базі правил

Наведені на рисунку 2.9 правила визначають керуючу команду для ресурсу BAND [44, 45]. Подібні правила розробляються для кожного ресурсу R_n . В кожному умову з бази правил замість змінних z'_1, z'_2, z'_n, q'_1 підставляються значення, які було визначено фазифікатором. Зі значень z'_1, z'_2, z'_n, q_1 , які є в одному правилі обирається найменше. Після цього дається співвідношення між нечітким поняттям «підвищити на багато, підвищити на середнє», «підвищити на мало», «нічого не робити», «зменшити на багато», «зменшити на середнє» та «зменшити на мало» та чітким.

Така операція виконується з кожним правилом з бази правил [46].

Після цього база правил отримує певну кількість значень для кожної лінгвістичної змінної «підвищити на багато», «підвищити на середнє», «підвищити на мало», «нічого не робити», «зменшити на багато», «зменшити на середнє» та «зменшити на мало» [43].

З кожного набору значень для лінгвістичних значень «підвищити на багато», «підвищити на середнє», «підвищити на мало», «нічого не робити»,

«зменшити на багато», «зменшити на середнє» та «зменшити на мало» обирається найменше та подаються на вихід бази правил. Отримані значення на рисунку 2.8 позначені літерами $f'_n, f''_n, f'''_n, \dots, f_n^{(7)}$.

Розглянемо детально фазу дефазифікації на прикладі одного ресурсу R_1 .

На вхід дефазифікатору надходять значення $f'_1, f''_1, f'''_1, \dots, f_1^{(7)}$ та виконується функція обернена фазифікації. Схема дефазифікатору для одного ресурсу R_1 наведена на рисунку 2.9. Фаза дефазифікації виконується по алгоритму Мамдані [42].

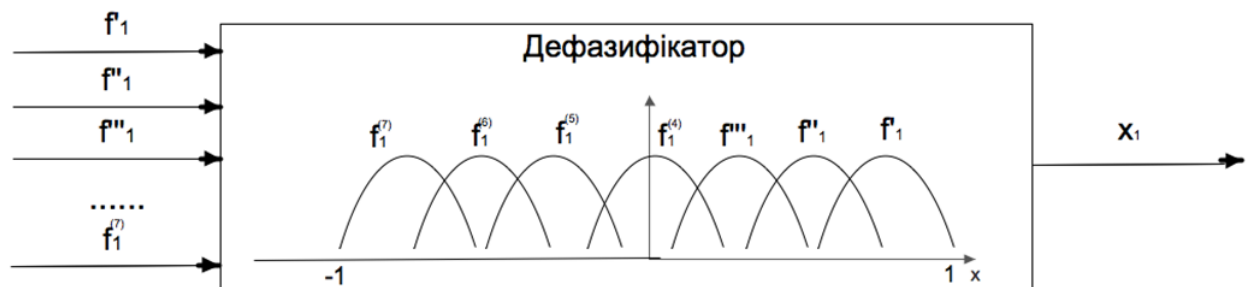


Рисунок 3.8 Фаза дефазифікації

На рисунку функції приналежності ресурсу до класу понижи на багато відповідає позначання $f_1^{(7)}$, до класу понизити на середнє $f_1^{(6)}$, до класу понижи на мало $f_1^{(5)}$, до класу нічого не робити $f_1^{(4)}$, до класу підвищити на мало f'''_1 , до класу підвищити на середнє f''_1 , до класу підвищити на багато f'_1 .

Кожне значення $f'_1, f''_1, f'''_1, \dots, f_1^{(7)}$ відкладається на вісь ОУ. Через знайдені точки проводяться прямі, паралельні осі ОХ. Далі будується фігура, крайніми точками якої буде точка перетину прямих зі своїми функціями приналежності.

На рисунку 2.10 представлено рисунок роботи алгоритму Мамдані для трьох значень нечітких змінних.

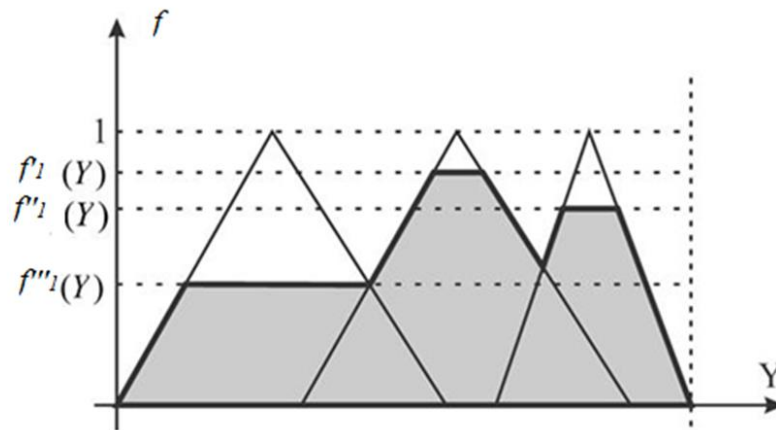


Рисунок 3.9 Приклад використання алгоритму нечіткого виведення
Мамдані

З фігури, яка зображена на рисунку 3.9 знаходиться центр мас, значення якого і буде рівним тому на скільки потрібно збільшити або зменшити ресурс.

Знайдене значення x_n поступає на вхід механізму, який вміє безпосередньо віддавати команди серверу про змінення об'єму наданих ресурсів [46].

Після вибору складових системи управління постає задача розробити структуру системи управління, якій присвячений наступний розділ.

3.5 Динамічний перерозподіл ресурсів

ІТ-інфраструктура постачальника ІТ-сервісів складаються з багатьох обчислювальних вузлів, вузлів зберігання та транспортних вузлів. Кожен вузол має ряд ІТ-ресурсів, таких як центральний процесор (ЦП), пам'ять випадкового доступу (RAM), фізична пам'ять, пропускна здатність мережі тощо.

Запропонований двоетапний метод забезпечує статистичну оцінку рівня якості ІТ-сервісу без будь-яких припущень щодо ймовірнісної характеристики процесів, що відбуваються в інфраструктурі постачальника ІТ-служб.

Оскільки кожна послуга надається декількома віртуальними машинами $V = \{V_{\eta\varpi_\eta o} \mid \varpi_\eta = \overline{1, \dots, \zeta_\eta}, o \in [1, O], \eta = \overline{1, \dots, H}\}$, то процес розрахунку рівня якості

послуги, що надає кожна з віртуальних машин доволі трудомісткий. Тому запропоновано використовувати непрямий метод оцінки якості послуг, що застосовує методи непараметричного аналізу і нечіткої логіка.

Суть методу полягає в тому, що для віртуальної машини, що надає забезпечує ІТ-послугу, на підставі накопиченої статистики встановлюється залежність значень показників якості від значень задіяних обсягів ресурсів. Управління рівнем послуг зводиться до того, що система управління постійно проводить визначення задіяних обсягів ресурсів кожної віртуальної машини, після чого система управління оцінює поточний рівень показника якості послуги, що надається віртуальною машиною і з урахуванням критерію (3.1) приймає рішення по управлінню рівнем послуг. При цьому використовується два механізми управління для підтримки рівня послуг на оговореному в SLA рівні. Один механізм заснований на зміні обсягу ресурсів, виділених для підтримки ІТ-послуги. У даній роботі здійснюється збільшення або зменшення кількості віртуальних машин, що надають окрему ІТ-послугу. Інший механізм використовує балансування навантаження на віртуальні машини. Для кожної ІТ-послуги ініціюється окремий балансувальник навантаження, який є складовою частиною системи управління.

Для кожної віртуальної машини $V_{j,i}$, $j=\overline{1,M_i}$, $i=\overline{1,K}$ визначається поточне значення якості ІТ-послуги, що вона надає. Для цього для кожної віртуальної машини визначається значення $r_{m,i}^*$, $m=\overline{1,L}$, $i=\overline{1,K}$ задіяних ресурсів.

Виходячи з величини $r_{m,i}^*$, $m=\overline{1,L}$, $i=\overline{1,K}$ і враховуючи вирази (6)–(8) розраховується значення $L_{b,i,\gamma}^q$, $b=\overline{1,D_i}$, $\gamma \in [1, \Gamma_{b,i}]$.

Отримані значення $L_{b,i,\gamma}^q$, підставляються в нечіткі правила (9) і визначається степінь приналежності $L_{i,\omega}^Q$, $\omega=\overline{1,\Omega_i}$.

Для знаходження числового значення рівня якості ІТ-послуги для кожної віртуальної машини Q_i , визначається центр мас за фігури (10), котра представляє

собою результат агрегації функція приналежності $\mu_{i,\omega}$, що обмежені згори значенням $L_{i,\omega}^Q$, що визначається за формулою (11).

Якщо для всіх віртуальних машин $V_{j,i}$, $j = \overline{1, M_i}$, $i = \overline{1, K}$ отримані значення рівня якості ІТ-послуги Q_i нижче узгодженого, то система управління приймає рішення про збільшення кількості віртуальних машин, що надають i -у послугу на 1.

Якщо для всіх віртуальних машин $V_{j,i}$, $j = \overline{1, M_i}$, $i = \overline{1, K}$ отримане значення якості послуги Q_i перевищує узгоджений рівень, то приймається рішення знизити споживання ресурсів серверів – зменшити кількість доступних віртуальних машин для i -го сервісу на 1.

Якщо отримане значення рівня якості послуги Q_i не знаходиться в межах норми, то балансувальник перерозподіляє запити користувачів між віртуальними машинами $V_{j,i}$, $j = \overline{1, M_i}$, $i = \overline{1, K}$ пропорційно значенням Q_i .

РОЗДІЛ 4 РОЗРОБЛЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ ЯКІСТЮ КОРПОРАТИВНИХ ІТ-СЕРВІСІВ

4.1 Проектування програмного продукту для визначення якості мультимедійних сервісів

При проектуванні архітектури застосунку був обраний компонентний підхід з розбивками на шари. Основна увага в даному типі архітектури приділяється розкладанню дизайну на окремі функціональні або логічні компоненти, що надають чітко визначені інтерфейси, що містять методи, події і властивості. В даному випадку забезпечується більш високий рівень абстракції і не відбувається концентрації уваги на таких питаннях, як протоколи зв'язку або загальний стан. Також було зроблено розбиття на шари, що забезпечує угруповання пов'язаної функціональності додатку в різних шарах, що вибудовуються вертикально, поверх один одного. Функціональність кожного шару об'єднана спільною роллю або відповідальністю. Шари слабо пов'язані, і між ними здійснюється явний обмін даними. Таке розбиття системи забезпечує гнучкість, а також зручність і простоту обслуговування.

При проектуванні системи приділялася увага тому, щоб система мала наступні якості:

- придатність для повторного використання в різних сценаріях і роботи з різними сервісами;
- можливість заміщення дозволяє оновити окремий модуль без оновлення всієї системи;
- незалежність від контексту – компоненти проектуються для роботи з різними сервісами;
- інкапсуляція – компоненти надають інтерфейси, що дозволяють стороннім компонентам використовувати їх функціональність, не

розкриваючи при цьому деталі внутрішніх процесів або внутрішні змінні чи стан;

- внутрішня незалежність – компоненти проєктуються з мінімальними залежностями від інших компонентів. Таким чином, компоненти можуть бути розгорнуті в будь-якій зручній середовищі без впливу на інші компоненти або системи.

Основні переваги обраного архітектурного стилю:

- простота розгортання – існуючі версії компонентів можуть замінюватися новими сумісними версіями, не впливаючи на інші компоненти або систему в цілому;
- простота розробки – для забезпечення заданої функціональності компоненти реалізують широко відомі інтерфейси, що дозволяє вести розробку без впливу на інші частини системи;
- спрощення з технічної точки зору – компоненти спрощують систему через використання контейнера компонентів і його сервісів.

Модуль оцінки якості роботи сервісів представлений такими елементами:

- шар доступу до даних складається з таких компонентів доступу до БД і компонента первинної обробки даних;
- бізнес шар складається з представлення бізнес-суті математичної моделі, суті статистики та сутності аналізу даних;
- шар уявлення складається з компонентів UI і компонентів логіки подання;
- шар внутрішньої взаємодії. Для забезпечення взаємодії між шарами.

Загальна архітектура модуля наведена на риску 4.1.

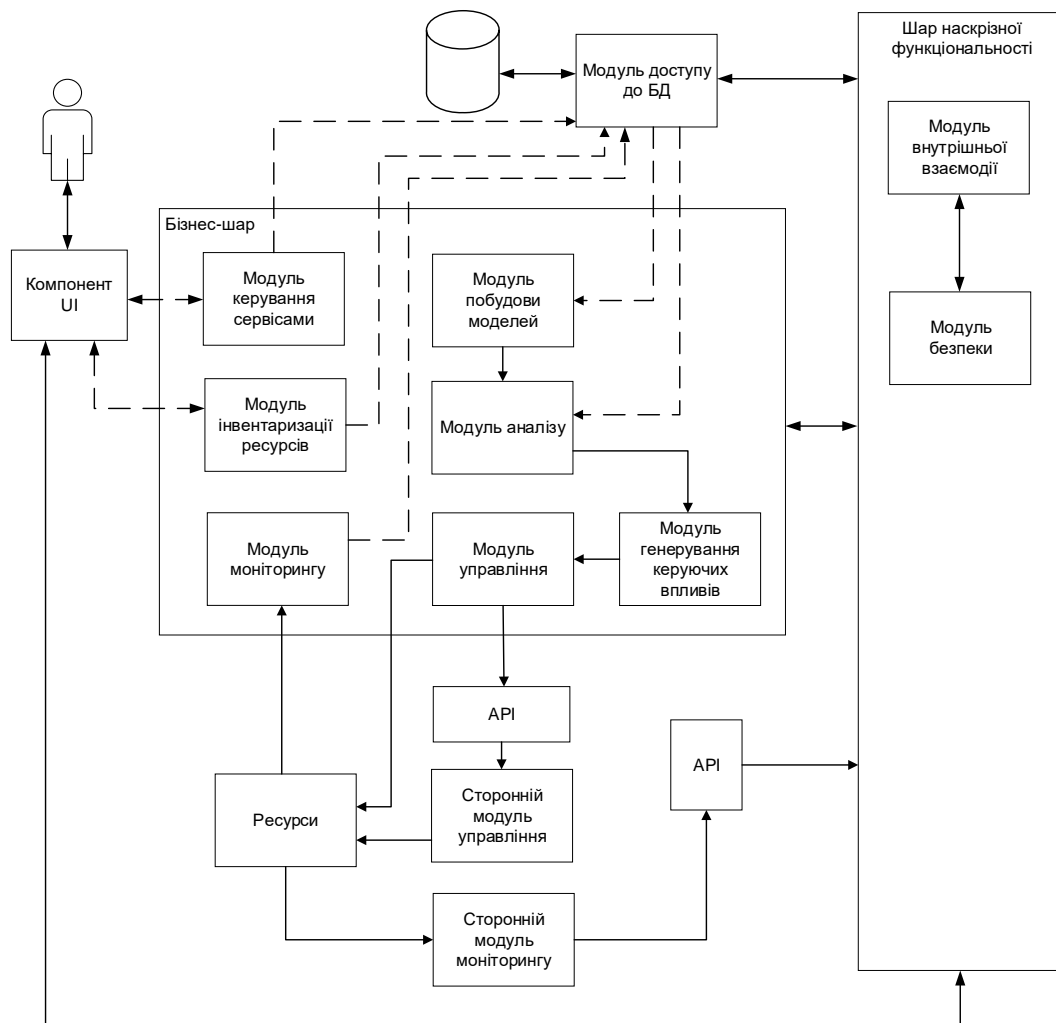


Рисунок 4.1 Загальна архітектура застосунку

Дані моніторингу мережі записуються в БД. Через заданий період часу дані моніторингу надходять на модуль оцінки якості, де по ним визначається якість функціонування обраного сервісу. Результат записується в БД. При подачі відповідного керуючого впливу, модель може переходити в режим донавчання. Формат повідомлення для керуючого впливу є XML-файл, в якому вказується або навчальна вибірка, або всі параметри роботи елементів ІТ-інфраструктури в даний момент часу

4.1.2 Шар доступу до даних

Шар доступу до даних включає наступні компоненти:

- компоненти доступу до даних. Ці компоненти абстрагують логіку, необхідну для доступу до базових сховищ даних. Вони забезпечують централізацію загальної функціональності доступу до даних, що сприяє спрощенню налаштування компонентів.
- компоненти первинної обробки даних. Ці компоненти забезпечують контроль правильного формату вхідних даних. При необхідності здійснюють нормування і обробку вхідних даних.
- модуль обробки керуючого впливу. Обробляє інформацію, що зберігається в файлі налаштувань. Шар доступу до даних повинен відповідати вимогам додатка, працювати ефективно і безпечно і забезпечувати простоту обслуговування і розширення в разі зміни бізнес-вимог. Проектування шару доступу до даних визначає наступні завдання:

Вимоги в шару:

- вибір технології доступу до даних. Вибір технології доступу до даних залежить від типу даних, з якими доведеться працювати, і того, як передбачається обробляти дані в додатку.
- інкапсуляція функціональності доступу до сховища даних. Шар доступу до даних повинен приховувати деталі доступу до джерела даних. Він повинен забезпечувати управління підключеннями, формування запитів і зіставлення сутностей додатки зі структурами джерела даних.
- зіставлення сутностей додатки зі структурами джерела даних. Тип сутності, використовуваної в додатку, є основним фактором при прийнятті рішення про метод зіставлення цих сутностей зі структурами джерела даних та обробці виключень, що виникають при обробці даних. Шар доступу до даних повинен перехоплювати і обробляти (або хоча б забезпечувати початковий етап) всі винятки, пов'язані з джерелами даних і операціями.

4.1.3 Бізнес шар

Бізнес шар включає в себе сутність математичної моделі. Бізнес-сутність математичної моделі (БСММ) інкапсулюють бізнес-логіку і дані, необхідні для подання в додатку об'єктів математичних моделей. Вони зберігають значення даних і надають їх через властивості, містять узагальнені бізнес-дані моделей і керують ними, надають програмний доступ до збереженням стану і пов'язаної функціональності. БСММ також перевіряють надходять чи на вхід сутності дані, а також дані які містяться в ній; вони інкапсулюють бізнес-логіку для забезпечення несуперечності даних, а також реалізації їх поведінки.

Сутність математичної моделі складається з наступних компонентів:

- модуль обробки вхідних даних. Даний модуль аналізує, обробляє і нормує вхідні дані необхідні для роботи конкретної математичної моделі. Також він відповідає за перетворення/переформатування даних;
- модуль-бібліотека математичних функцій. Модуль для зручного швидкого доступу до різних математичних функцій для побудови, навчання і тестування математичних моделей;
- модуль навчання. Модуль забезпечує процес навчання математичної моделі. Відповідає також за перенавчання моделей за новими даними, що надійшли. Такими даними, наприклад, може бути нова навчальна вибірка;
- модуль перевірки. Модуль контролює якість навчання математичної моделі і сигналізує про те, чи можливо її застосовувати в реальних умовах;
- модуль забезпечення основного режиму роботи. Контролює роботу математичної моделі після її навчання і перевірки.

Компонента роботи зі статистикою входить до модулю забезпечення основного режиму роботи. Бізнес-сутність роботи зі статистикою (БСПС) інкапсулюють бізнес-логіку і дані, необхідні для подання в додатку об'єктів

даних статистики. Вони зберігають значення даних і надають їх через властивості, містять узагальнені бізнес-дані статистики і керують ними. БСРС також перевіряють надходять на вхід суті дані, а також дані які містяться в ній; вони інкапсулюють бізнес-логіку для забезпечення несуперечності даних, а також реалізації їх поведінки.

Компонента роботи зі статистикою складається з наступних компонентів:

- модуль збору статистичних даних. Даний модуль забезпечує логування і збір статистичних даних;
- модуль обробки даних для зберігання. Даний модуль готує дані для зберігання, має елементи управління для швидкого доступу до них;
- сутність аналізу даних. Бізнес-сутність аналізу даних (БСАД) інкапсулюють бізнес-логіку і дані, необхідні для подання в додатку об'єктів аналізу даних. Вони зберігають значення даних і надають їх через властивості, містять узагальнені бізнес-дані аналізу і керують ними. БСАД також перевіряють чи надходять на вхід дані, а також самі дані; вони інкапсулюють бізнес-логіку для забезпечення несуперечності даних, а також реалізації їх поведінки.

Компонента роботи з математичними моделями складається з наступних компонентів:

- модуль обробки вхідних даних. Даний модуль аналізує, обробляє і нормує вхідні дані необхідні для роботи конкретної математичної моделі. Також він відповідає за перетворення/переформатування даних;
- модуль-бібліотека математичних функцій. Модуль для зручного швидкого доступу до різних математичних функцій для побудови, навчання і тестування математичних моделей;
- модуль навчання. Модуль забезпечує процес навчання математичної моделі. Відповідає також за перенавчання моделей за новими надійшли даними. Такими даними, наприклад, може бути trouble-tickets;

- модуль перевірки. Модуль контролює якість навчання математичної моделі і сигналізує про те, чи можливо її застосовувати в реальних умовах;
- модуль забезпечення основного режиму роботи. Контролює роботу математичної моделі після її навчання і перевірки.

Компонента роботи зі статистикою. Бізнес-сутності роботи зі статистикою (БСРС) інкапсулюють бізнес-логіку і дані, необхідні для подання в додатку об'єктів даних статистики. Вони зберігають значення даних і надають їх через властивості, містять узагальнені бізнес-дані статистики і керують ними. БСРС також перевіряють надходять на вхід суті дані, а також дані які містяться в ній; вони інкапсулюють бізнес-логіку для забезпечення несуперечності даних, а також реалізації їх поведінки.

Компонента роботи зі статистикою складається з наступних компонентів:

- модуль збору статистичних даних. Даний модуль забезпечує логування і збір статистичних даних;
- модуль обробки даних для зберігання. Даний модуль готує дані для зберігання, має елементи управління для швидкого доступу до них.

Компонента аналізу даних. Бізнес-сутності аналізу даних (БСАД) інкапсулюють бізнес-логіку і дані, необхідні для подання в додатку об'єктів аналізу даних. Вони зберігають значення даних і надають їх через властивості, містять узагальнені бізнес-дані аналізу і керують ними. БСАД також перевіряють дані, що надходять надходять; вони інкапсулюють бізнес-логіку для забезпечення несуперечності даних, а також реалізації їх поведінки.

Компонента роботи зі статистикою складається з модулю аналізу оброблених даних. Даний модуль має доступ до статистичних даних, а також великий математичний апарат їх аналізу і обробки.

Шар представлення включає наступні компоненти:

- компоненти для користувача інтерфейсу. Це візуальні елементи;
- додатки, що використовуються для відображення даних користувачеві і прийому призначеного для користувача введення;

- компоненти логіки подання. Логіка уявлення - це код додатка, який визначає поведінку і структуру програми, але не залежить від конкретної реалізації призначеного для користувача інтерфейсу.

Одною з основних задач, при проектуванні шару представлення є поділ функціональної області. При проектуванні будуть використовуватися спеціальні компоненти UI для формування візуального представлення, відображення і взаємодії з користувачем. Також застосування в даному шарі спеціальних сутностей представлення дозволило представляти бізнес-логіку і дані в формі, зручній для використання компонентами UI і логіки подання. Компоненти шару подання інкапсулюють бізнес-логіку і дані бізнес-шару в рамках шару подання і використовують їх.

4.1.4 Шар наскрізної функціональності

Шар наскрізної функціональності забезпечує загальна функціональність програми, яку не можна віднести до конкретного шару або рівню. Даний шар складається з наступних компонентів:

- модуль безпеки. Забезпечує безпеку передачі даних між шарами. Обробляє виняткові ситуації і забезпечує максимальну стабільність роботи програми.

4.1.5 Модуль внутрішньої взаємодії

Модуль внутрішньої взаємодії забезпечує зв'язок між шарами. Надає верствам і сутностям всі необхідні дані.

Було розроблено та побудовано стенд для дослідів. Він використовувався для проведення експериментів для виявлення залежності якості сервісу HTTP від кількості наданих ресурсів.

Експерименти проводилися з різною кількістю користувачів, які генерувалися оператором. Ресурси, вплив яких досліджувався для виявлення залежності між кількістю споживаних ресурсів на якість сервісу: оперативна

пам'ять, пропускна здатність каналу зв'язку, кількість процесорного часу та кількість користувачів, які використовували сервер в момент часу t .

Під час дослідження виявлено, що для досліджуваного сервісу використовувати більше двох ядер не є раціональним. Також виявлено, що кількість оперативної пам'яті не сильно впливає на час завантаження веб-сторінки для досліджуваного сервісу.

В ході експерименту зроблено висновок, що бажано перерозподіляти такий ресурс як пропускна здатність каналу зв'язку, оскільки зміна цього ресурсу не вимагає перезавантаження серверу та процент доступності процесору віртуального серверу. Зміна кількості процесорів вимагає перезавантаження системи.

4.2 Розроблення стенду для проведення дослідження середовища надання ІТ-послуг HTTP сервісу

Даний розділ присвячений проведенню дослідження середовища надання ІТ-послуг для виявлення зв'язку між якістю ІТ-сервісу та кількістю задіяних ресурсів. Ці дослідження необхідні для навчання штучної нейронної мережі самостійно оцінювати якість наданої послуги, шляхом аналізу об'єму споживаних ресурсів

Наступні досліді проводяться для того, щоб дізнатися про вплив таких ресурсів як оперативна пам'ять (ОЗУ), кількість наданих ядер, процесорний час (подається у відсотках) та полоси пропускання мережевого інтерфейсу.

Схема стенду наведена на рисунку 4.2.

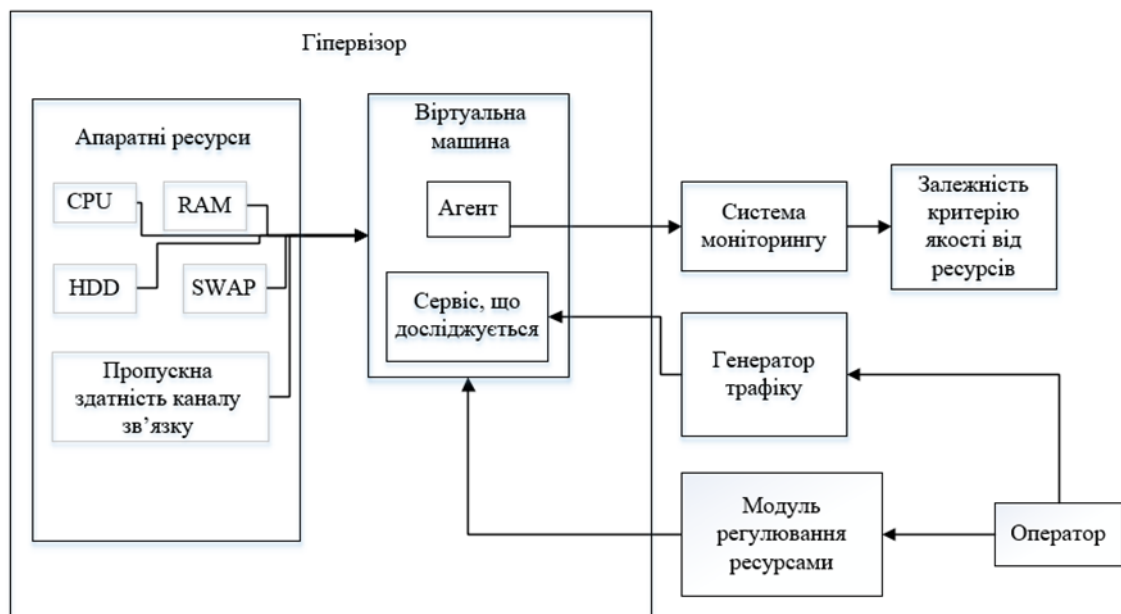


Рисунок 4.2 Схема стенду для експериментів

Стенд складається з гіпервізору, системи моніторингу, генератору трафіку та модуля регулювання ресурсами.

На самому гіпервізорі налаштовується віртуальна машина, якій надається набір деяких ресурсів. Кількість кожного з ресурсів можна надавати або забирати в будь-який момент часу, але деякі з них потребують перезавантаження системи для того щоб зміни вступили в силу. Також кількість наданих ресурсів можна змінювати або з гіпервізору, або з самої віртуальної системи.

Кількість віртуальних машин обмежується потужністю гіпервізору та ресурсами, які є в нього.

На віртуальній машині налаштовується агент, який періодично знімає дані щодо споживаних ресурсів віртуальною машиною. Даний агент відсилає дані на систему моніторингу, який збирає всі отримані значення, компонує та відображає статистику в зручному для оператора вигляді. Система моніторингу має критерій, за яким вона оцінює якість досліджуваного сервісу. Наприклад, час відгуку веб-сторінки. Якщо даний критерій не задовільний, тобто більший за заданий, то система моніторингу оповіщає оператора про це оператора. СМ демонструє про те, яке було споживання ресурсів в кожен момент часу, тому

можна визначити при якому наборі ресурсів якість досліджуваного сервісу була незадовільна.

Генератор трафіку потрібен для того, щоб надсилати запити зі збоку користувача на досліджуваний сервіс. Даний модуль може регулювати кількість одночасних запитів, їх зміст, періодичність, тривалість.

Задача оператора складається в тому щоб запускати генератор трафіка з його набором параметрів та дивлячись на якість отриманого сервісу, надавати та забирати ресурси у віртуальної машини.

Для проведення експериментів було обрано систему Linux дистрибутиву Debian7, на якій налаштована KVM-віртуалізація. На даній ОС налаштований віртуальний сервер з операційною системою Linux. Для дослідів було обрано веб-сервіс Apache2 та програмне забезпечення Atlassian Jira, написане на мові Java. В якості системи моніторингу було обрано програмне забезпечення Zabbix. Генератор трафіку обрано програмне забезпечення Apache Jmeter. Ресурсами оператор керує за допомогою командного рядку.

Для чистоти експерименту дослід буде проводитися по схемі “один сервер – один сервіс”. Також було завершено всі зайві процеси та проводилися експерименти тоді, коли мережа не була надто перевантажена.

Ресурси, якими буде керувати оператор: кількість ядер, процесорний час, кількість оперативної пам’яті та пропускна здатність каналу зв’язку.

Кількість віртуальних ядер, які буде використовувати віртуальний сервер, регулюється з командного рядка гіпервізору. Даний ресурс вимагає перезавантаження системи для того, щоб зміни вступили в силу.

Процесорний час, який надається для виконання обраного процесу регулюється з командного рядка системи на якій встановлено сервіс. Даний ресурс можна змінювати в реальному часі, він не вимагає перезавантаження системи.

Кількість оперативної пам’яті, які буде використовувати ВМ, регулюється з командного рядка гіпервізору. Даний ресурс можна змінювати в реальному часі.

Полоса пропускання каналу зв'язку регулюється з системи, на якій встановлене програмне забезпечення. Даний ресурс можна змінювати в реальному часі.

Технічні характеристики гіпервізору наведено в таблиці 4.1.

Таблиця 4.1 – Технічні характеристики серверу на якому виконується віртуалізація

Кількість процесорів	8
Технічні дані процесора	Intel(R) Xeon(TM) CPU 3.00GHz
Оперативна пам'ять	8Gb
Операційна система	Debian GNU/Linux 7
Розрядність системи	x86_64
Тип віртуалізації	KVM

4.2.2 Досліди з участю одного користувача

Для подальших експериментів з'ясовувалося скільки ядер процесору потрібно для того, щоб сервіс функціонував без негативного впливу на якість. Проводячи експеримент було надано для досліджуваного процесу якийсь відсоток процесорного часу. Кожне ядро може надати 100% свого часу. Тобто 8 ядер можуть надати 800% усього процесорного часу.

Досліди проводилися для 1, 2, 3 та 4 ядер. На рисунку 4.2 показані графіки залежності часу завантаження веб-сторінки, який позначається t_{zc} і вимірюється в секундах, від кількості наданого процесорного часу, який позначається P_n і вимірюється в відсотках.

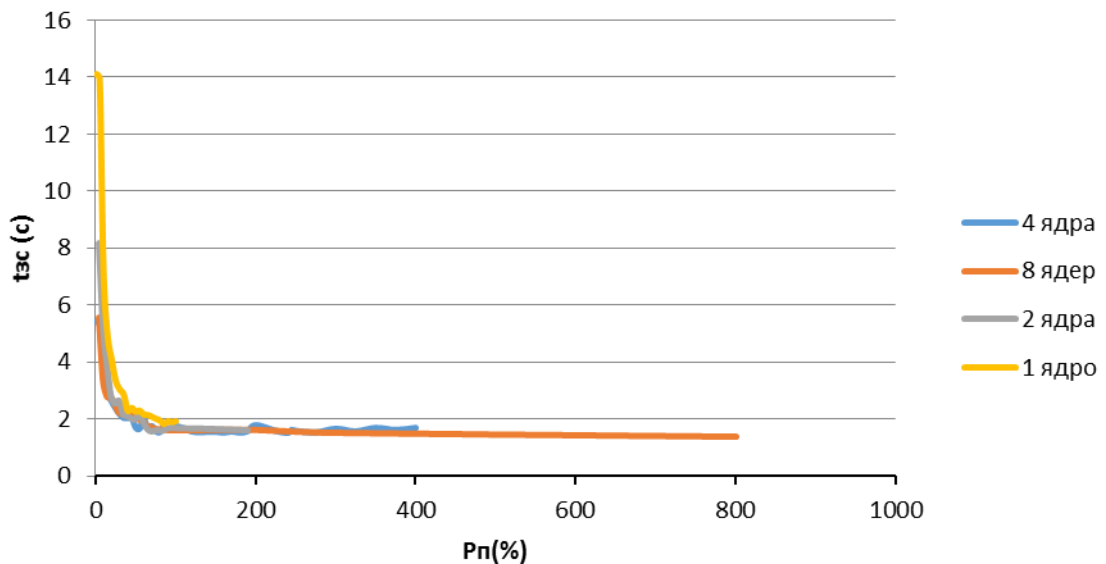


Рисунок 4.3 Графік залежності часу завантаження веб-сторінки від наданого процесорного часу

Як видно з рисунку 4.2 залежність має експоненціальний характер. На рисунку 4.3 наведено графік 4.2 у збільшеному масштабі в інтервалі від 0 до 100% процесорного часу графік залежності часу завантаження веб-сторінки від кількості наданого процесорного часу. Як видно з рисунку нижче, при виділенні 1 та 2 ядер, час завантаження при 10% процесорного часу більший за 6 секунд. При виділенні 4 та 8 ядер графіки повністю співпали. З цього можна зробити висновок, що для досліджуваного процесу виділяти більше, чим 4 ядра немає необхідності.

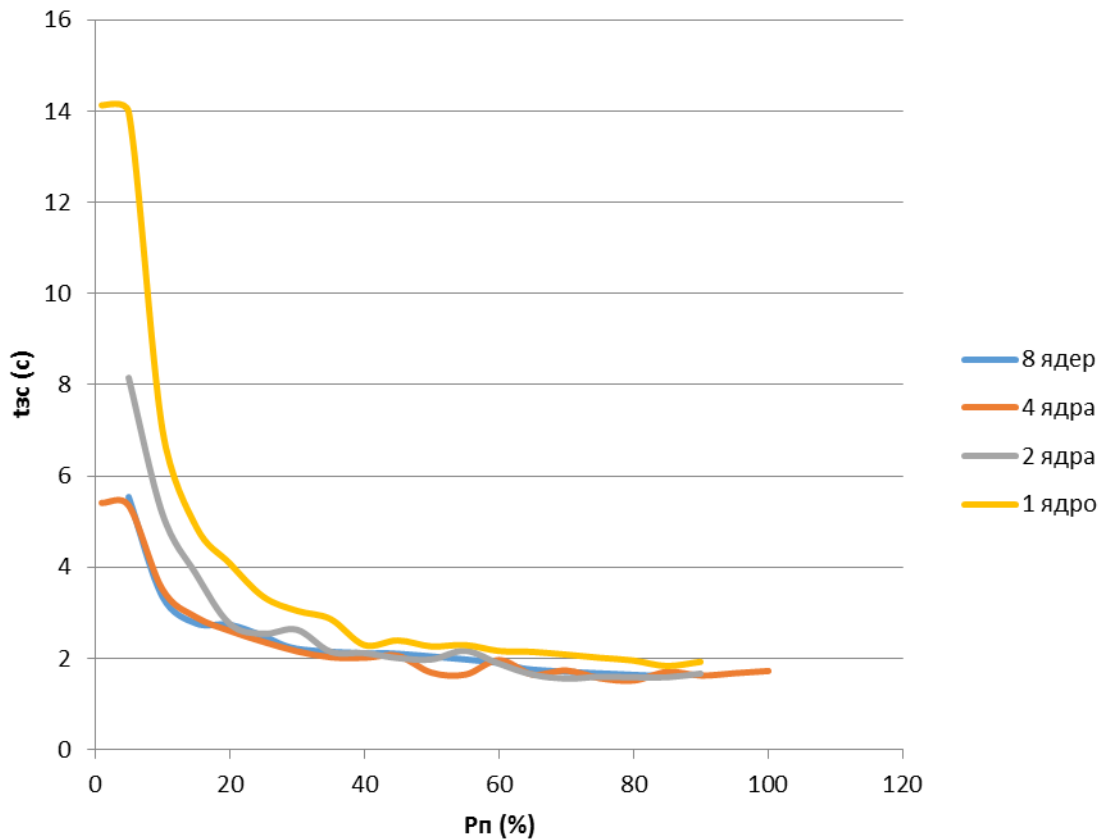


Рисунок 4.4 Графік залежності часу від наданого процесорного часу у збільшеному масштабі

Для дослідження впливу оперативної пам'яті та кількості ядер на початкове завантаження сторінки було проведено наступний експеримент. Вперше веб-сторінка завантажується більше ніж 15 секунд. Це пояснюється тим, що завантажена сторінка кладеться до оперативної пам'яті, щоб в наступному разі http-сторінка швидше завантажувалася у користувача. Це дуже схоже на роботу кешу. Досліди проводилися для 2, 4, 6 та 8 ядер. На рисунку 4.4 наведено графік часу завантаження веб-сторінки, який позначений як t_{zc} та вимірюється в секундах, від кількості наданої оперативної пам'яті, яка позначена ОЗУ та вимірюється в Мб.

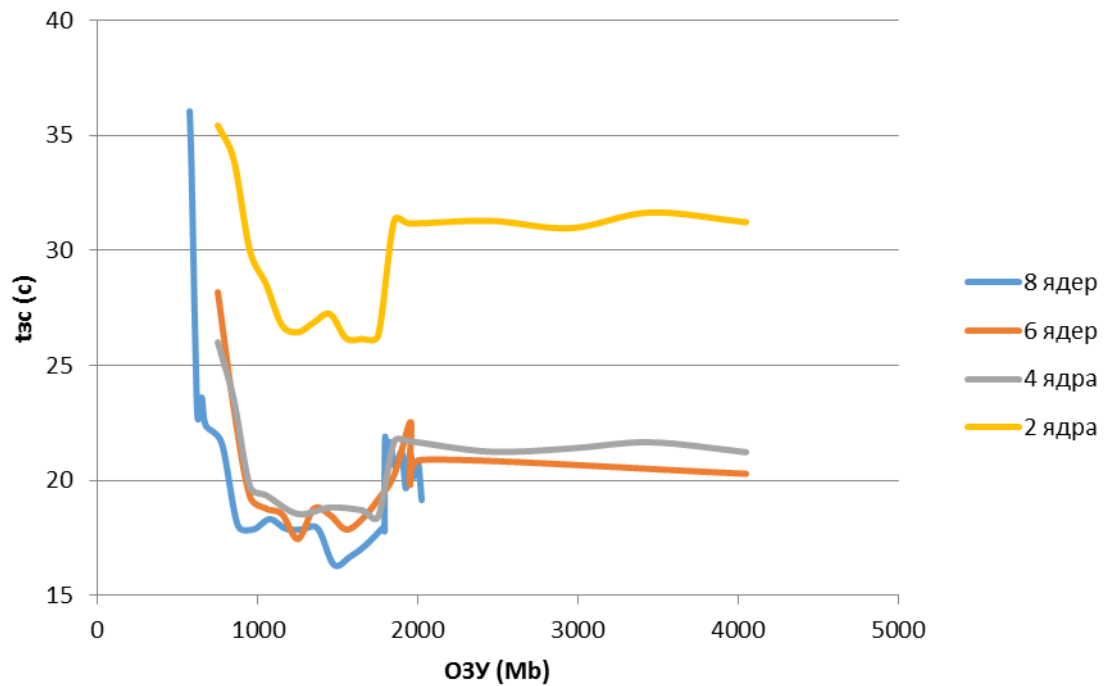


Рисунок 4.5 Графік залежності часу завантаження веб-сторінки вперше від наданої оперативної пам'яті

Як видно з поданого вище графіку при використанні 2 ядер час завантаження значно гірший, ніж при 4, 6 та 8 ядрах. При кількості ядер більше 4, графіки майже співпадають. Перегин в інтервалі від 1000 до 2000 Мб пояснюється тим, що при таких ресурсах в оперативну пам'ять кладеться менший об'єм даних та використовується менший об'єм процесорного часу.

Для виявлення впливу кількості оперативної пам'яті на час завантаження веб-сторінки було проведено наступний дослід. На рисунку 4.5 наведений графік впливу виділеної оперативної пам'яті, яка позначається як ОЗУ і вимірюється в Мб на вторинне завантаження веб-сторінки, яке вимірюється в секундах та позначається $t_{зс}$.

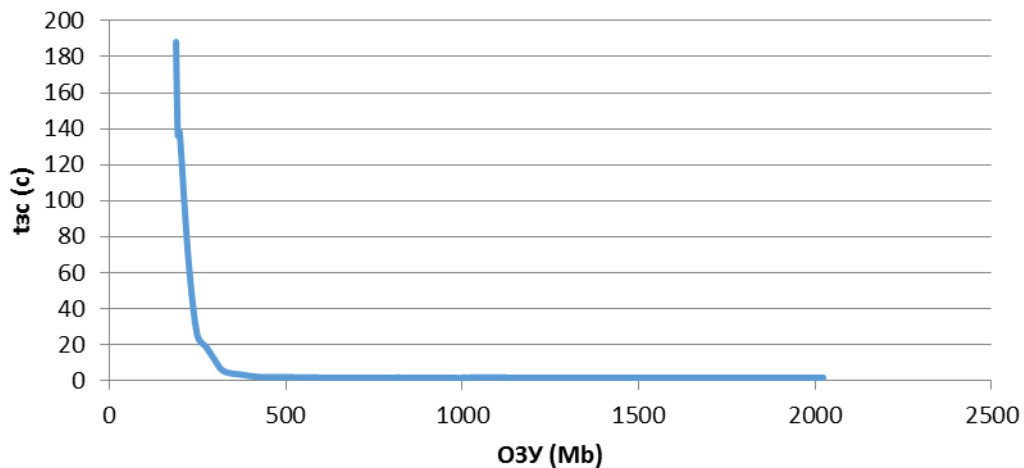


Рисунок 4.6 Графік залежності часу завантаження веб-сторінки від наданої оперативної пам'яті

Як видно з графіку вище якість сервісу має експоненціальну залежність. На рисунку 4.6 наведено графік у збільшеному масштабі залежності часу завантаження веб-сторінки від наданої оперативної пам'яті. Збільшений графік лежить в інтервалі від 300 до 600 Мб.

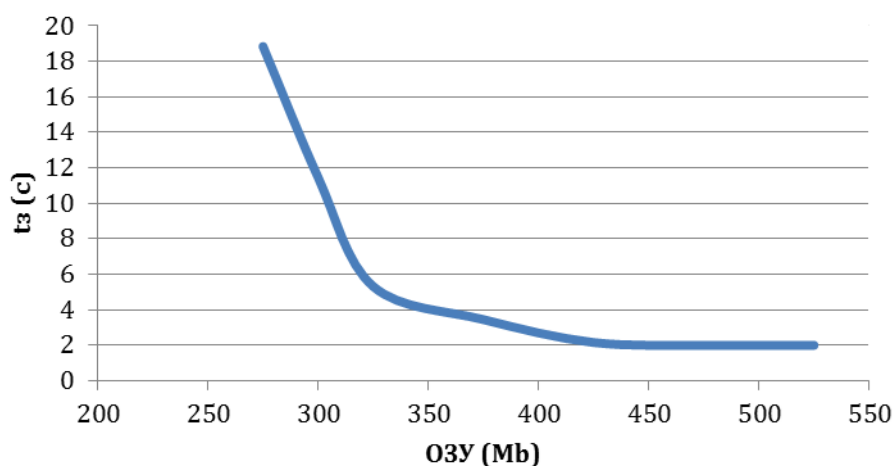


Рисунок 4.7 Графік залежності часу завантаження веб-сторінки від наданої оперативної пам'яті у збільшеному масштабі

З вище наведеного графіку можна помітити, що якість сервісу погіршується при 400 Мб пам'яті. Але при таких ресурсах сам сервіс вже не

перезапуститься та система не завантажується. Тому в даному досліді оперативна пам'ять не має великий вплив на якість наданого сервісу.

Для виявлення характеру залежності часу завантаження веб-сторінки від виділеної пропускної здатності каналу зв'язку було проведено наступний дослід. На рисунку 4.7 наведений графік залежності часу завантаження веб-сторінки, який позначений як t_{zs} та вимірюється в секундах, від пропускної здатності каналу зв'язку, який позначається як ПЗ та вимірюється в kbit/s.

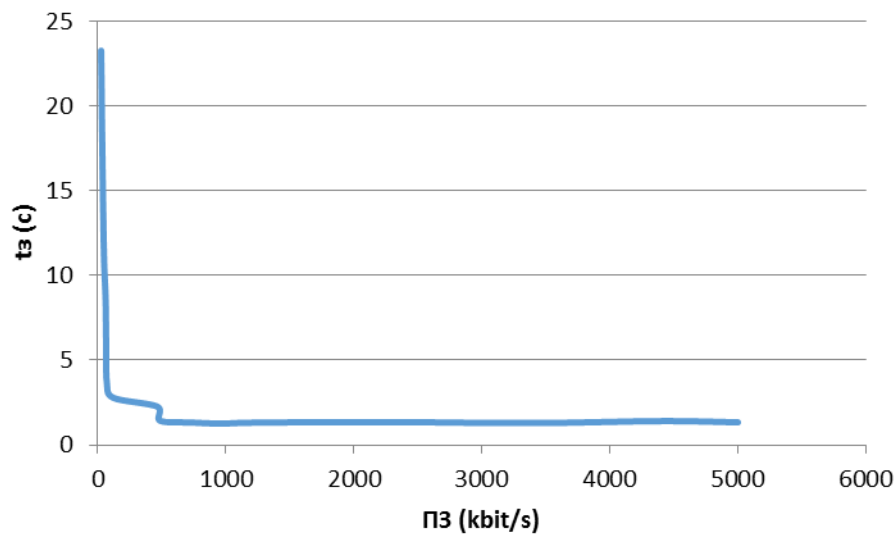


Рисунок 4.8 Графік впливу пропускної здатності каналу зв'язку на час завантаження веб-сторінки

Як видно з графіку залежність має експоненціальний характер. На рисунку 4.8 показаний графік 4.7 у збільшеному масштабі в інтервалі від 0 до 500 kbit/s.

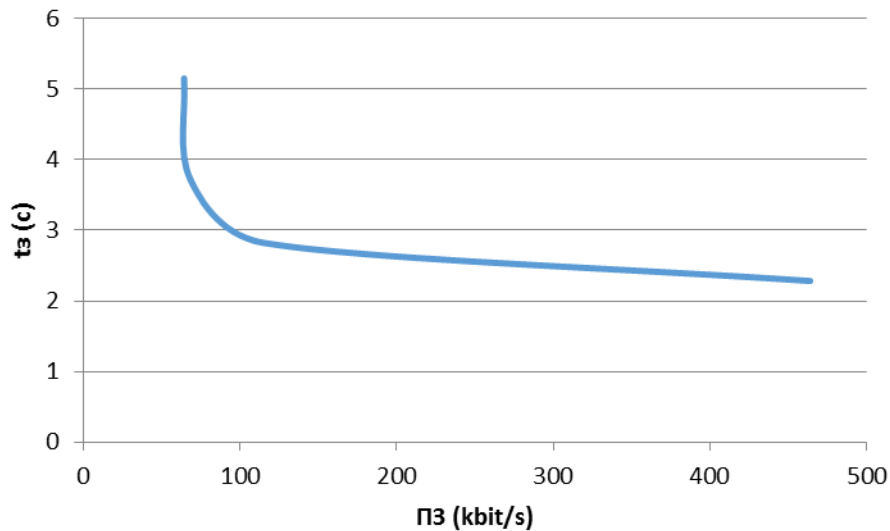


Рисунок 4.9 Графік залежності часу завантаження веб-сторінки від пропускної здатності каналу зв'язку у збільшеному масштабі

Як видно з графіку якість сервісу падає при наданні пропускної здатності каналу зв'язку менше 100кбит/s.

З попередніх дослідів було виявлено, що оперативна пам'ять не впливає на якість сервісу, тому даним ресурсом було знехтувано. В таблиці 4.2 наведено результати експерименту. В даному досліді брали участь такі ресурси: пропускна здатність каналу зв'язку (ПЗ) та процесорна потужність (P_n). ПЗ вимірюються в кбит/s, а P_n вимірюється у відсотках. В тілі таблиці наведено значення часу завантаження веб-сторінки, який вимірюється у секундах. Зеленим кольором виділені значення часу завантаження сторінки, який було визначено як «відмінний», жовтий та червоний – «задовільний» та «незадовільний» відповідно.

Таблиця 4.2 – Вплив процесорної потужності (ПП) та пропускної здатності (ПЗ) каналу зв'язку на якість HTTP сервісу

ПП	ПЗ										
	5300	3400	2300	1200	443	220	106	81	59	35	25
400		1.4	1.41	1.42			1.7	1.8	6.37	14.4	25.6
80			1.63	1.66		1.9	2.01	2.04	6.54		

70			1.69	1.7	1.97	2.03	2.1	2.17	6.83		
60		1.75	1.87	1.77	2.06	2.12	2.16	2.2	7.11		23.7
50	1.88	1.94	1.89	1.9	2.1	2.13	2.2	2.26	8.2	17.7	
40	2.35	2.05	2.02	2.06	2.2	2.24	2.25	2.3		18.6	
30	2.65	2.33	2.25	2.34	2.26	2.47	2.51	3.1		19.7	31.9
10	5.46	4.89	4.67	5.11	5.36	5.72	5.75	5.9			
5	11.9				11	10.6	11.6	11.7	11.8		

Експерименти з деякими набором ресурсів не було проведено, бо їх значення не впливали на результати експерименту.

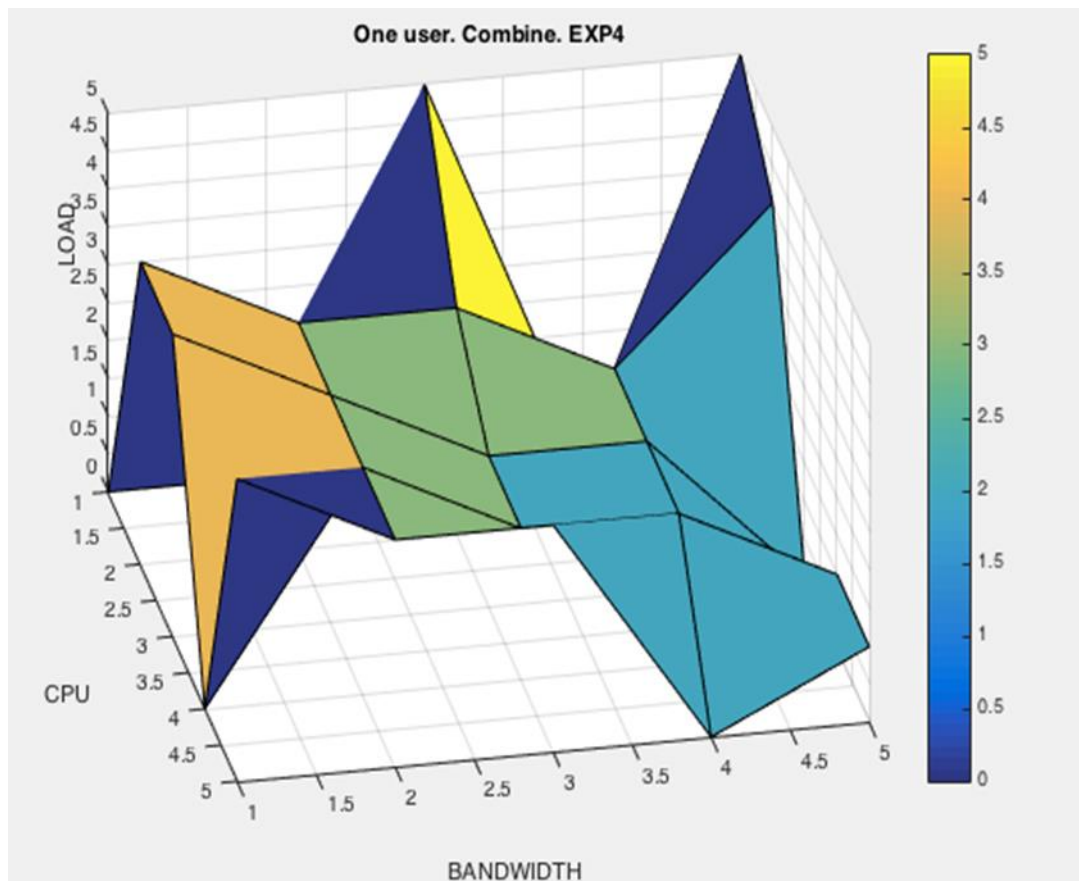


Рисунок 4.10 Графік залежності часу завантаження веб-сторінки від наданих ресурсів пропускної здатності та процесорної потужності

На рисунку 4.10 наведено тривимірний графік, який ілюструє при яких значеннях ресурсів якість сервісу добра, задовільна та погана. На поданому

нижче рисунку пропускна здатність каналу зв'язку позначена як Bandwidth, процесорна потужність – CPU, та часу завантаження веб-сторінки – LOAD. Значення параметрів пропускної здатності, процесорної потужності та часу завантаження сторінки подані у перерахованому масштабі в логарифмічній шкалі з натуральним логарифмом по основі 4.

4.2.3 Досліди з великою кількістю користувачів

Так як будь-яким сервісом буде користуватися більше, ніж один користувач, було проведено серію експериментів, щоб виявити вплив кількості користувачів на якість досліджуваного сервісу.

Для проведення дослідів було обрано програму Apache Jmeter.

В ході експерименту на сервіс буде впливати навантаження в 1, 5, 15, 25, 50, 100 користувачів.

У розділі 2.4 уговорювалося, що час завантаження веб-сторінки розділяється на три групи – відмінний, задовільний та незадовільний. Час завантаження, який лежить в інтервалі від 3 до 6 секунд вважається задовільним та у таблицях буде позначений жовтим кольором. Менший час, тобто менше 3 секунд вважається відмінним та позначається зеленим кольором, більший за 6 секунд – незадовільний та позначається червоним кольором.

Для дослідження впливу процесорного часу на час завантаження веб-сторінки виділяється 4 ядра. Вище було виявлено, що 4 ядра повністю задовольняють поставлене завдання. Більша кількість не є доцільною, так як вони не використовуються на повну потужність, а менше – істотно впливають на якість послуги, що надається.

Даний експеримент проводиться зі зниженням доступності ресурсів процесора (у відсотковому співвідношенні) від 400 до 10. У ході проведення експерименту так само було виявлено, що при кількості користувачів більше 50 час завантаження сторінки значно більше значення, яке визначається як «задовільний». Нижче наведена таблиця 4.3 з результатами даного експерименту.

Кількість користувачів вказана у лівому стовпчику, потужність процесора позначена як P_n , вказана у нижньому рядку та вимірюється у відсотках. В тілі таблиці вказаний час завантаження веб-сторінки, який вимірюється у секундах. Зеленим кольором виділені значення часу завантаження сторінки, який було визначено як «відмінний», жовтий та червоний – «задовільний» та «незадовільний» відповідно.

Таблиця 4.3 – Залежність часу завантаження сторінки від P_n для різної кількості користувачів при використанні

Кількість користувачів	P_n , %								
	400	300	250	200	150	100	50	25	10
1	1.17	1.08	1.12	1.42	1.12	1.35	1.37	2.7	3.15
5	1.67	1.82	1.5	2.8	2.2	2.15	2.52	5.81	13.29
15	2.35	2.24	2.39	3.9	2.42	3.36	3.44	12.22	30.42
25	2.41	2.55	2.54	2.74	3.23	4.43	4.48	19.95	50
50	3.65	5.76	4.5	5.27	5.47	5.2	5.73	31.94	62

На рисунку 4.10 представлений тривимірний графік, який ілюструє при яких ресурсах якість сервісу добра, при яких – ні. Кількість користувачів позначена як Users, час завантаження веб-сторінки позначений як LOAD та вимірюється у секундах, надана процесорна потужність позначена як CPU та вимірюється у відсотках.

Даний графік зображений у логарифмічній шкалі з натуральним логарифмом по основі 4.

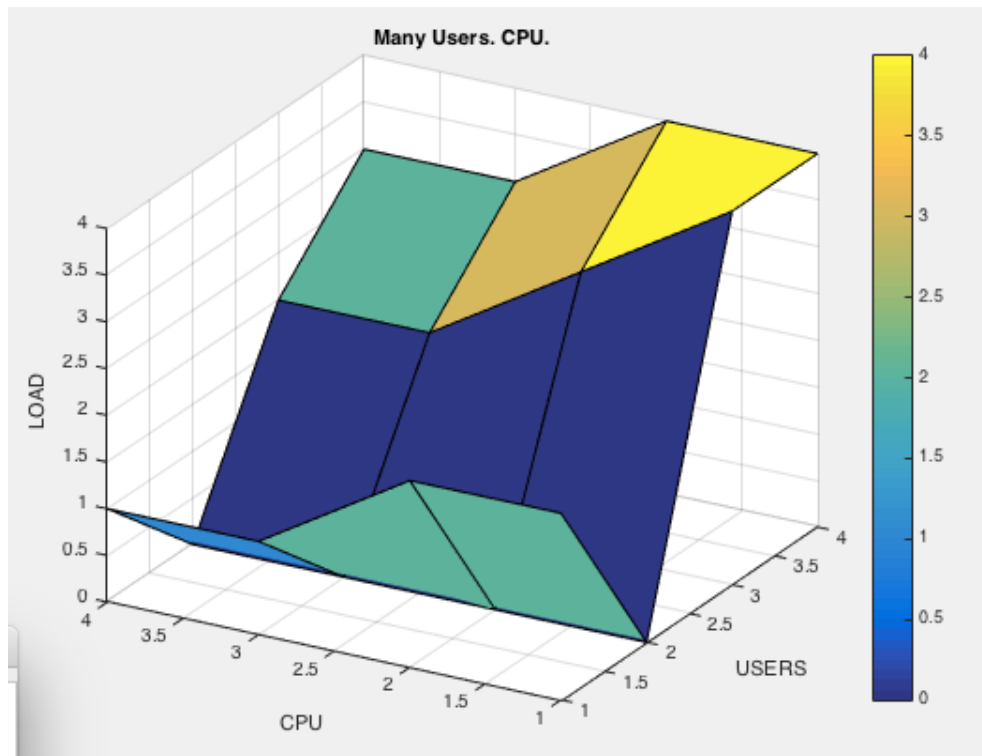


Рисунок 4.1 – Залежність часу завантаження сторінки від кількості наданого процесорного часу та кількості користувачів

Для виявлення впливу кількості оперативної пам'яті та кількості користувачів на якість сервісу, що досліджується було проведеному наступний дослід.

В таблиці 4.4 наведені результати експерименту. Кількість користувачів зазначена у крайньому лівому стовпчику, кількість виділеної оперативної пам'яті, яка позначена як ОЗУ та вимірюється в Мб зазначена у нижньому рядку. В тілі таблиці зазначений час завантаження веб-сторінки, який вимірюється у секундах. Час, який належить до класу «відмінно» позначений зеленим кольором, а «задовільний» – жовтим.

Таблиця 4.4 – Вплив кількості оперативної пам'яті на якість

Кількість користувачів	ОЗУ, Мб			
	2003	1855	1655	1155
1	1.16	1.05	1.07	1.18
5	1.67	1.9	2	2

15	2.56	2.95	3.68	2.8
25	2.41	3.8	3.4	3.4
50	3.65	5.65	5.78	5.84

Як видно з вище приведеної таблиці оперативна пам'ять на дану кількість користувачів не має великий вплив. Тому надалі в експериментах цим параметром буде знехтувано.

Для виявлення впливу полоси пропускання каналу зв'язку на якість сервісу, що надається було проведено наступний експеримент.

В таблиці 4.5 наведені результати дослідів. Зеленим кольором виділені значення часу завантаження сторінки, який було визначено як «відмінний», жовтий та червоний – «задовільний» та «незадовільний» відповідно.

Кількість користувачів зазначена у крайньому лівому стовпчику, кількість виділеної полоси, яка позначена як ПП та вимірюється в kbit/s зазначена у нижньому рядку. В тілі таблиці зазначений час завантаження веб-сторінки, який вимірюється у секундах.

Таблиця 4.5 – Залежність часу завантаження веб-сторінки від полоси пропускання та кількості користувачів

Кількість користувачів	Полоса пропускання, kbit/s					
	3400	2300	1100	580	235	118
1	1.17	1.17	1.29	1.88	3.8	7.44

Продовження таблиці 4.5 – Залежність часу завантаження веб-сторінки від полоси пропускання та кількості користувачів

Кількість користувачів	Полоса пропускання, kbit/s					
	3400	2300	1100	580	235	118
5	1.67	2.38	3.61	5.84	16	27.3
15	2.35	4	12.26	12.85	35.6	88
25	2.42	5.39	16.38	19.19	86	

50	3.65	9.9	25.77	33.53		
----	------	-----	-------	-------	--	--

Для виявлення впливу комбінованих ресурсів на якість сервісу, що надається було проведено наступний експеримент. Було взято такі ресурси: пропускна здатність каналу зв'язку, яка вимірюється в kbit/s та процесорний час, який вимірюється у відсотках. Результати проведених експериментів наведені в таблицях 4.6 – 4.15. Оскільки на час завантаження веб-сторінки впливає три параметри, трьохвимірний графік побудувати не вдалося. Таблиці було згруповано за наданим процесорним часом.

Кількість користувачів зазначена у крайньому лівому стовпчику, кількість виділеної полоси пропускання зазначена у верхньому рядку. В тілі таблиці зазначений час завантаження веб-сторінки, який вимірюється у секундах. Час, який належить до класу «відмінно» позначений зеленим кольором, а «задовільний» – жовтим, «незадовільний» – червоним.

Таблиця 4.6 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 400% процесорного

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118
100	10.71	16.89					
50	3.7	8.56		16.96			
25	2.41	5.76	6.78	9.44	22.3		
15	2.35	3.94	4.6	6.48	12.34	31.46	
5	1.5	1.5	2.9	3.53	5.98	13.55	25.4
1	1.15	1.23	1.22	1.4	2	3.86	7.47

Таблиця 4.7 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 300% процесорного часу

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118

50	6.34	9.24					
25	4.9	5.43	6.86				
15	2.8	3.55	4.63	7.55	12.12		
5	1.8	2.25	2.53	3.73	5.85	14.02	
1	1.14	1.24	1.25	1.32	1.93	3.77	7.44

Таблиця 4.8 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 250% процесорного часу

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118
50	6.98	9.53					
25	3.04	5.62	7.04	11.4			
15	2.92	3.55	4.9	6.56	13.18		
5	2.14	2.24	2.63	3.85	5.94	12.75	
1	1.13	1.15	1.2	1.92	1.95	3.79	7.39

Таблиця 4.9 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 200% процесорного часу

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118
50	7.9	10.02					
25	3.67	5.7	7.11	11.23			
15	2.55	3.85	5.3	6.79	12.94		
5	2.03	2.3	2.38	3.58	5.87	13.6	
1	1.18	1.23	1.24	1.4	1.94	3.8	7.34

Таблиця 4.10 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 150% процесорного часу

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118

50	7	9.6					
25	3.9	5.17	6.75	9.84			
15	3.06	3.06	4.69	6.47	13.93		
5	1.9	2.36	2.8	3.27	5.85	13.74	
1	1.12	1.13	1.23	1.37	2	3.81	7.45

Таблиця 4.11 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 100% процесорного часу

Кількість користувачів	Смуга пропускання						
	3400	2300	1700	1100	580	235	118
50	6.51	9.64					
25	3.89	5.41	7.18	9.94			
15	2.92	4	4.6	6.71	11.92		
5	2	2.25	2.48	3.18	6	13.02	
1	1.11	1.18	1.23	1.42	2.06	3.82	7.38

Таблиця 4.12 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 50% процесорного часу

Кількість користувачів	Смуга пропускання					
	3400	2300	1100	580	235	118
25	6.92	7.18	11.04			
15	5.9	6.49	6.67	12.13		
5	2.93	3.25	3.2	5.94	13.43	
1	1.79	1.81	1.84	1.93	3.74	7.36

Таблиця 4.13 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 35% процесорного часу

Кількість користувачів						
	3400	2300	1100	580	235	118
15	10.56					

5	4.54	4.74	4.76	5.79		
1	1.95	1.88	1.96	2.07	3.76	7.35

Таблиця 4.14 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 25% процесорного часу

Кількість користувачів	Смуга пропускання					
	3400	2300	1100	580	235	118
5	6.34	7.03	7.18	7.41	13.84	
1	2.59	2.59	2.65	2.67	4	7.27

Таблиця 4.15 – Залежність ЧЗ від кількості користувачів, смуги пропускання та виділеного 10% процесорного часу

Кількість користувачів	Смуга пропускання
	3400
1	6.95

Дані експерименти було проведено так детально, щоб навчити нейромережу самостійно визначати якість сервісу в залежності від ресурсів, що використовує сервіс.

4.2.4 Оцінка якості функціонування сервісу після управління

Час завантаження веб-сторінки, який є еквівалентним якості досліджуваного сервісу, в момент часу t представлено на рисунку 4.10.

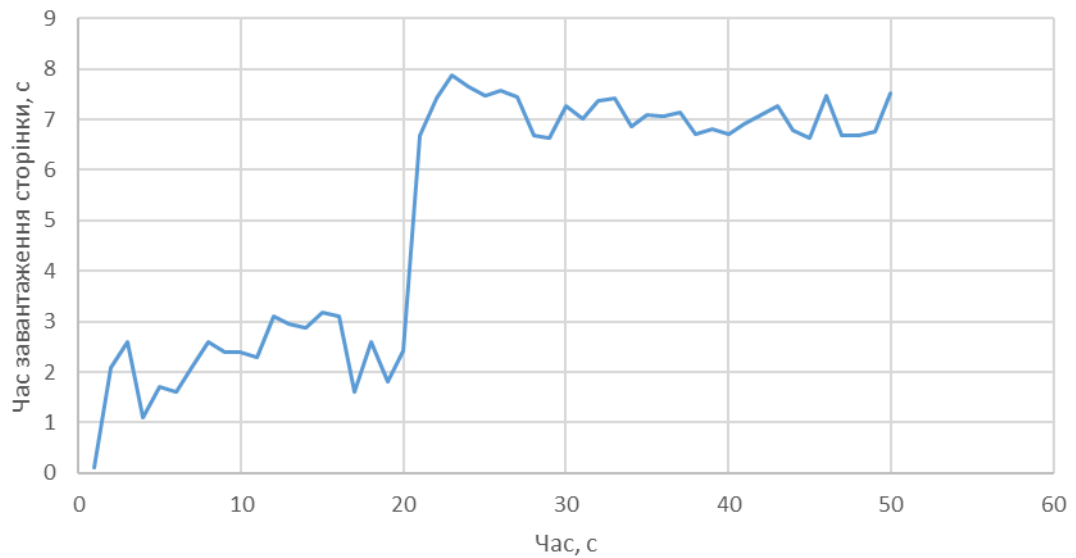


Рисунок 4.11 Час завантаження веб-сторінки без системи управління

Після проведення експериментів та навчання нейромережі, система встановлюється на сервер.

На рисунку 4.11 представлено графік, який відображає час завантаження веб-сторінки в момент часу t .

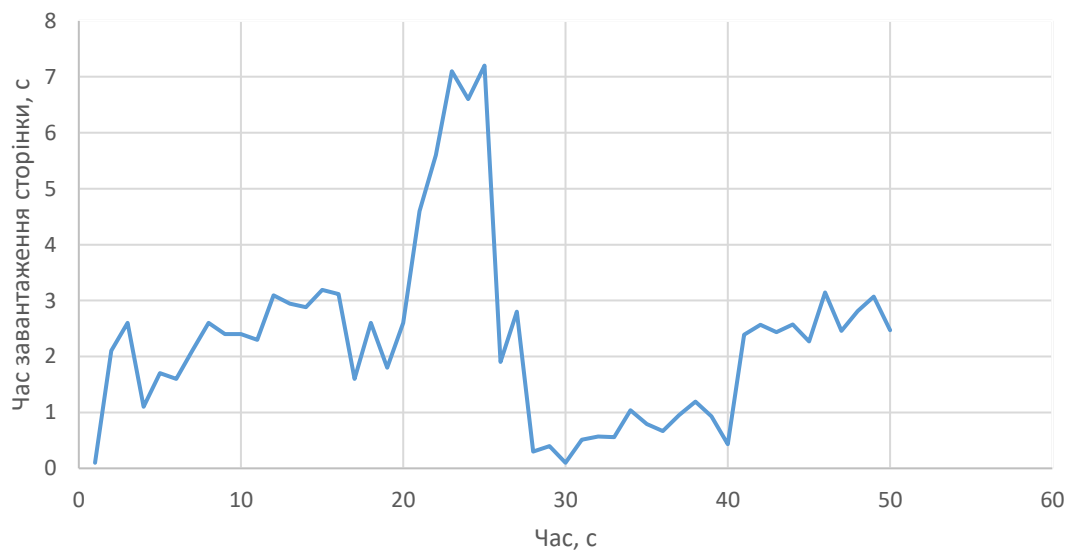


Рисунок 4.12 Час завантаження веб-сторінки в момент часу t

Як видно з графіку 4.12 в момент часу 25 секунд час завантаження веб-сторінки значно збільшився. Це пояснюється тим, що кількість користувачів досліджуваного сервісу збільшилася.

Кількість користувачів в момент часу t показана на рисунку 4.13.

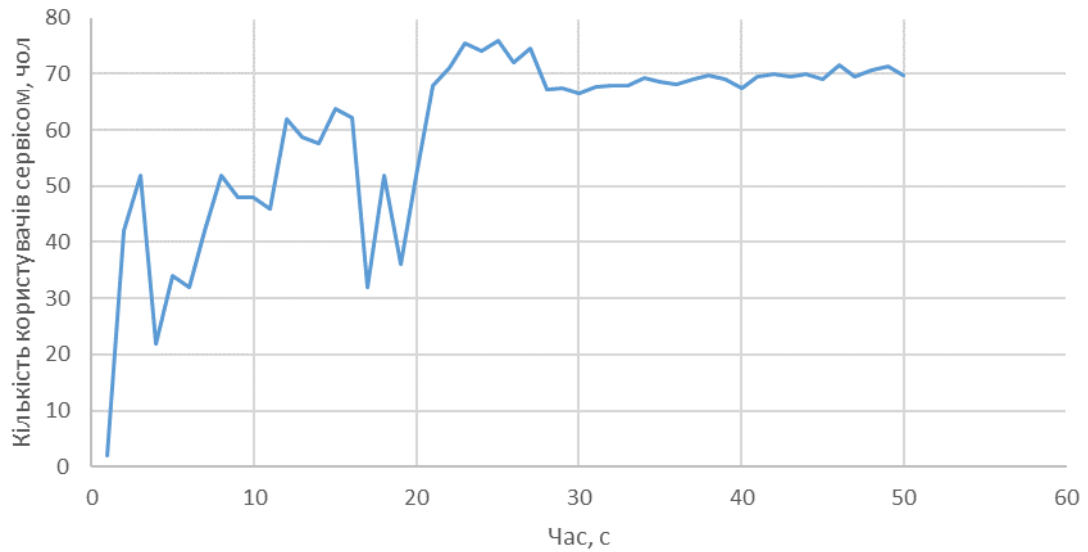


Рисунок 4.13 Кількість користувачів сервісу в момент часу t

Система управління після погіршення якості сервісу прийняла рішення про збільшення кількості об'єму ресурсів, а саме була створена додаткова віртуальна машина

Як видно з графіку 4.13, в момент часу 20 секунд кількість користувачів почала стрімко зростати, тому якість сервісу HTTP стрімко погіршала. Після того, як управління було виконано та кількість ресурсів для веб сервера збільшена, якість сервісу стала відмінною, тому система управління віддала команду про зменшення кількості ресурсів, доки якість сервісу не стала задовільною.

ВИСНОВКИ

У дисертаційній роботі, присвяченій вирішенню важливої науково-практичної проблеми розробки комплексу математичних моделей, методів, алгоритмів і інформаційної технології, аналізу та управління рівнем якості корпоративних ІТ-сервісів:

1. На основі аналізу проблем управління якістю ІТ-послуг у корпоративній ІТ-інфраструктурі встановлено, що для підвищення конкурентоспроможності та ефективного виконання бізнес-процесів постачальників ІТ-послуг необхідно підтримувати якість ІТ-послуг на узгодженому рівні за умов раціонального використання ресурсів ІТ-інфраструктури.

2. Визначено, що рівнем якості ІТ-послуг, в умовах консолідації та віртуалізації обчислювальних ресурсів, можна керувати шляхом регулювання обсягів обчислювальних ресурсів ІТ-інфраструктури, які виділені застосуванням, що надають ці послуги. Підтримка рівня якості надаваних ІТ-інфраструктурою на узгодженому рівні, вимагає розробку моделей та методів аналізу якості ІТ-послуг та створення відповідної інформаційної технології і системи управління якістю сервісів для комплексного вирішення задач управління ІТ-інфраструктурою.

3. Запропоновано метод оцінювання поточного рівня якості ІТ-послуг з використанням апаратів нечіткої логіки та нейронних мереж. Метод дозволяє на основі даних моніторингу обладнання та подальшого зведення метрик до рівня сервісів визначати поточний рівень якості ІТ-послуг та ступень задоволеності користувачів даними послугами.

4. Запропоновано метод структурного навчання нейронних мереж, який здатний динамічно підлаштовувати структуру нейронної мережі безпосередньо під час її навчання. Даний метод дозволив зменшити помилку нейромережевих класифікаторів та автоматично підлаштовувати структуру нейромережевого класифікатору відповідно до поставлених задач управління якістю ІТ-послуг.

5. Запропоновано метод оцінювання поточного рівня якості сервісу, що надається ІТ-інфраструктурою, з використанням апаратів нечіткої логіки та

непараметричної статистики. Метод дозволяє на основі даних моніторингу параметрів компонент ІТ-інфраструктури визначати компоненти ІТ-інфраструктури, функціонування яких має відхилення від нормативного.

6. Запропоновано метод оцінювання якості ІТ-послуг з використанням апаратів нечіткої логіки та непараметричної статистики, що дозволяє визначати якість послуги, що надавалася протягом заданого інтервалу часу, на основі поточних значень параметрів функціонування.

7. Запропоновано метод управління якістю ІТ-послуг на основі даних моніторингу роботи інформаційно-телекомунікаційних ресурсів ІТ-інфраструктури. Метод дозволяє генерувати керуючі впливи на ІТ-інфраструктуру при систематичному відхиленні рівня якості послуги від узгодженого рівня.

8. Запропоновано підхід до налаштування системи автоматичного управління ІТ-інфраструктурами. Даний підхід використовує апарат нечіткої логіки для визначення керуючих впливів. Сценарії, на які має реагувати система автоматичного управління прописуються у формі нечіткої бази правил. При відхиленні рівня якості надаваного сервісу, система автоматично визначає сукупність сценаріїв, що необхідно задіяти та ресурси ІТ-інфраструктури, що треба розподілити між її елементами для забезпечення узгодженого рівня якості сервісів.

9. Розроблено метод балансування навантаження на сервери ІТ-інфраструктури, що дозволяє розподіляти запити користувачів на сервери відповідно до поточного рівня якості ІТ-послуг, що задають дані сервери. Розроблений метод дозволяє запобігати зменшенню рівня якості надаваних ІТ-послуг відповідно до поточного навантаження на сервери ІТ-інфраструктури.

10. Розроблені моделі, методи та алгоритми використані при проектуванні і реалізації підсистеми управління інфраструктурами ТОВ «СІТІУС ПРО», ТОВ «АЙАДМІН». Це дозволило скоротити в середньому на 15-20% час на проектування і реалізацію підсистем, скоротити витрати на експлуатацію серверного парку на 10-15% з одночасним збільшенням ефективності його використання. На основі моделей, методів та інформаційних технологій

управління IT-інфраструктурами розроблено методичне забезпечення, яке використане в навчальному процесі кафедри автоматики та управління в технічних системах КПІ ім. Ігоря Сікорського.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Rolik O., Kolesnik V., Halushko D. Decomposition-Compensation Method for IT Service Management. In: Kulczycki P., Kóczy L., Mesiar R., Kacprzyk J. (eds) Information Technology and Computational Physics. CITCEP 2016. Advances in Intelligent Systems and Computing, vol 462, Springer, Cham, 2017, pp. 89-107.
2. Telenyk S. Models and methods of resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko // Technical transaction. Automatic control. – Politechnica Krakowska, 2013. – vol. 4-AC. – pp. 41–52
3. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // Information Technology, Computational and Experimental Physics. Kulczycki P., Kowalski P. A., Łukasik S. (eds). – AGN-UST. – 2016. – pp. 13-16.
4. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: ВЕК+, 2010. – № 52. – С. 39–52
5. Ролик А.И. Оценка качества предоставления мультимедийных сервисов с использованием нейросетевого классификатора / Ролик А.И., Галушко Д.А., Барна В.В., Томащук А.В., Ясочка М.В. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Століття +, 2015. – № 63. – С. 25–30.
6. Ролик А.И. Метод нечеткой непараметрической оценки качества функционирования элементов информационно-телекоммуникационных систем / А.И. Ролик, Д.А. Галушко, П.Ф. Можаровский, А.А. Андриенко, В.М. Вовк // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2012.– Вип. 19 (39).– С. 115–125.
7. Structural Optimization of Neural Network for Qualitative Evaluation Method of IT-Infrastructure Functioning / Yaroslav Yu. Dorogyy, Sergii F. Telenik, Dmytro A. Halushko, Vasyl V. Tsurkan // Information and telecommunication sciences

: international research journal. – 2015. – Vol. 6, N. 2(11). – Pp. 36–43. – Bibliogr.: 15 ref

8. Теленик С.Ф. Система управления ИТ-инфраструктурой – путь к повышению эффективности функционирования предприятия / С.Ф. Теленик, А.И. Ролик, М.М. Букасов, А.В. Волошин, Д.А. Галушко // Інформаційні технології – інструмент підвищення конкурентоздатності підприємств: 9–10 груд. 2009.: матеріали конф. – К.: УкрНЦ РІТ, 2009. – С. 30–33.

9. Ролик А.И. Метод оценки состояния элементов информационно-телекоммуникационных систем на основе нечетких нейронных сетей/ А.И. Ролик, Д.А. Галушко, Ю.А. Кононенко // Обчислювальний інтелект (Результати, проблеми, перспективи): матеріали II-ї Міжнар. наук.-техн. конф. «Обчислювальний інтелект (QI-2013)», м. Черкаси, 14-18 травня 2013 р. – Черкаси: – 2013. – С. 233

10. Telenyk S. Data center resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko// : матеріали конф Proc. of the International Congress on Control and Information Processing 2013 (ICCIP'13) 7–8 December, Cracow, Poland. – 2013.

11. Ролік О.І. Метод оцінки стану елементів інформаційно-телекомунікаційних систем на базі апарату нечіткої логіки / О.І. Ролік, Д.О. Галушко, І.М. Плехова, Б.В. Ступак // : матеріали конф Physical and technological problems of radio engineering devices, telecommunication, nano- and microelectronics. Proceedings of the 3th international scientific-practical conference. Chernivtsi, Ukraine, October 24–26, 2013. – Чернівці: КОД. – 2013. – С. 104–105.

12. Qualitative evaluation method of IT-infrastructure elements functioning / S. Telenyk, O. Rolick, M. Bukasov, Y. Dorogiy, D. Halushko, A. Pysarenko // : матеріали конф 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Odessa, 2014, pp. 165-169.

13. Ролик А.И. Метод оценки качества телекоммуникационных сервисов на основе нейронных сетей со структурным обучением / А.И. Ролик, Д.А. Галушко, Д.С. Захаров, А.В. Томащук// Обчислювальний інтелект (результати, проблеми, перспективи): III-я Міжнародна наук.-практ. конф., 12-15 травня 2015

р. Київ-Черкаси: матеріали. – Черкаси: видавець Чабаненко Ю., 2015. – С. 250—251.

14. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // : матеріали конф Proc. of the Congress on Information Technology, Computational and Experimental Physics 2013 (CITCEP'15) 18–20 December, Cracow, Poland. – 2015. – р. 176–179.

15. Ролик А.И. / А.И. Ролик, Д.А. Галушко, Н.В. Кравчун, Т.В. Кравченко // : матеріали конф ICACIT-2015: 3-я Міжнар. конф. з автоматичного управління та інформаційних технологій, 11-13 грудня 2015 р. м. Київ: матеріали. – К., 2015. – с. 80–83.

16. Rolik O. Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm / O. Rolik, V. Kolesnik and D. Halushko// Proc. of the International Conference Radio Electronics & Info Communications (UkrMiCo). – Kiev, Ukraine. – 2016. – pp. 1–5.

17. D. Halushko, O. Rolik, and V. Samoty, “A Load Balancing Mechanism Based on Fuzzy Nonparametric Analysis of QoS Parameters” in Proc. CLOUD COMPUTING 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, February 19-23, 2017 Athens, Greece, IARIA, 2017, матеріали конф; pp. 102-107.

18. Abu M. T. Osman, “On the direct product of fuzzy subgroups”, Fuzzy Sets and Systems, vol. 12, pp. 87—91, 1984.

19. IT Service Quality Management Based on Fuzzy Logic O Rolik, V Kolesnik, D Halushko International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018 – Proceedings 31 January 2019, матеріали конф, Pages 604-608

20. Ролик А.И. Управление корпоративной ИТ-инфраструктурой / А.И. Ролик, С.Ф. Теленик, М.В. Ясочка // К.: Наукова думка, 2018. – 576 с.

21. ISO/IEC 20000-1 Information technology. Service management. Part 1: Specification. ISO/IEC, 2005. — 16 p.

22. ISO/IEC 20000-1:2005 Information technology. Service management. Part 2: Code of practice. ISO/IEC, 2005. — 34 p.
23. ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів (ISO/IEC 25010:2011, IDT) [Чинний від 2018-01-01], 2016. — 5 ст
24. ДСТУ ITU-T E.800:2014 Якість телекомунікаційних послуг. Терміни та визначення понять, 2015. — 5 р
25. ISO/IEC 9126:2001 Software engineering — Product quality — Part 1: Quality model, 2001. — 25 p
26. ДСТУ КО 9000:2015 Системи управління якістю. Існoвні положення та словник термінів, 2015 — 44 с
27. ДСТУ ISO 9001:2015. Системи управління якістю. Вимоги, 2015 — 30 с
28. Kearney, K.T.; Torelli, F. (2011). "The SLA Model". In Wieder, P.; Butler, J.M.; Theilmann, W.; Yahyapour, R. Service Level Agreements for Cloud Computing. Springer Science+Business Media, LLC. pp. 43–68. ISBN 9781461416142.
29. Ролик А.И. Тенденции и перспективы развития управления информационными технологиями / А.И. Ролик // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. — К.: «БЕК+», 2012. — № 55. — С. 81–109.
30. Теленик С.Ф. Зведення метрик оцінювання рівня обслуговування користувачів на основі експертних оцінок / С.Ф. Теленик, О.І. Ролік, О.М. Моргаль, О.С. Квітко// Вісник Вінницького політехнічного інституту. — 2011.— №1.— С. 112–123.
31. SLA Management Handbook. Vol. 1. Executive Overview. Ver. 2.1. TeleManagement Forum. — 2005. — 39 p.; Vol. 2. Concepts and Principles. Release 2.5. — TeleManagement Forum. — 2005. — 218 p.; Vol. 3. Service and Technology Examples. Ver. 2.1. — TeleManagement Forum. — 2005. — 89 p.

32. Boutaba R. Cyber Planner: A comprehensive toolkit for network service providers / R. Boutaba, J. Xiao, J. Aib — IEEE, 2008. — P. 382—386.
33. Cai Z., Chen Y., Kumar V., Milojicic D / Automated Availability Management Driven by Business Policies // 10th IFIP/IEEE Symposium on Integrated Management (IM). — 2007. — P. 264—273.
34. T. Chen, Y. Zhang, and X. Wang, G. B. Giannakis “Robust geographical load balancing for sustainable data centers,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3526—3530, 2016.
35. TSG-SA Working Group 1 // QoS Performance requirements for UMTS. — 1999 – 13 c.
36. J. Duan, “A data center virtualization framework towards load balancing and multi-tenancy,” IEEE 17th International Conference on High Performance Switching and Routing (HPSR), pp. 14—21, 2016.
37. A. Kumar and M. Kalra, “Load balancing in cloud data center using modified active monitoring load balancer,” International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), pp. 266—270, 2016
38. Boutaba R. CyberPlanner: A Comprehensive toolkit for network service Providers / R. Boutaba, J. Xiao, I. Aib // NOMS 2008 — 11th IEEE/IFIP Network Operations and Management Symposium. — 2008.— Vol. 11, No. 1. — pp. 379—386.
39. ISO/IEC 7498-1 Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model. Second editions. ISO/IEC, 1994. — 68 p.
40. Principles for a telecommunications management network: ITU-T Rec. M.3010. — Geneva. — 2000. — 36 p.
41. Катышев С. Об одной концепции управления распределенным ресурсами // Открытые системы. — 1998. — №3.
42. ISO/IEC 20000-1:2005 Information technology. Service management. Part 1: Specification. ISO/IEC, 2005. — 16 p.
43. ISO/IEC 20000-1:2005 Information technology. Service management. Part 2: Code of practice. ISO/IEC, 2005. — 34 p.
44. Josyula V. Cloud Computing. Automating the Virtualized Data Center. CiscoPress. / V.Josyula, M. Orr, G. Page. – Indianapolis. – 2012. – 392 c.

45. Chaganti P. Xen Virtualization. A Fast and Practical Guide to Supporting Multiple Operating Systems with the Xen Hypervisor. Packt. // P. Chaganti . – Birmingham. – 2007. – 132 с.
46. Михеев М.О. Администрирование VMware vSphere 5 / М. Михеев. – Москва. – 2012. – 504 с.
47. Методы робастного, нейро-нечеткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 744 с.
48. Вороновский Г.К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев // Харьков, ОСНОВА. – 1997. – 107 с.
49. Круглов В.В. Искусственные нейронные сети. Теория и практика. 2-е изд. // В.В. Круглов, В.В. Борисов // ГЛ Телеком. – 2002. – 382 с.
50. M. Joachimczak Artificial metamorphosis: Evolutionary design of transforming, soft-bodied robots/ M. Joachimczak, R Suzuki, T Arita // Artificial Life (2016) 22 (3): P. 271–298
51. Bongard, J. C., & Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In F. Hara & R. Pfeifer (Eds.), Morpho-functional Machines: The new species. Tokyo: Springer., P. 237–258
52. Dellaert, F., & Beer, R. D. (1996). A developmental model for the evolution of complete autonomous agents. In From animals to animats 4: Proceedings of the 4th International Conference on Simulation of Adaptive Behavior (SAB 1996) (pp. 393–401). Cambridge, MA: MIT Press.
53. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы // Д. Рутковская, М. Пилиньский, Л. Рутковский. – ГЛ Телеком. – 2006. – 256 с.
54. A Serdar Tasan A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries / A Serdar Tasan, Mitsuo Gen // Computers & Industrial Engineering №3(62), 2012, P. 755-761

55. Букасов М. М. Технологія управління ІТ-інфраструктурою на основі ресурсного підходу / Букасов М. М., Ролік О. І., Теленик С. Ф. // Вісник ЖДТУ.— 2008.— №4(47). — с. 180—189.
56. Хайкин С. Нейронные сети. Полный курс. Второе издание // С. Хайкин. — Вильямс. — 2006. — 1104 с.
57. Zadeh L.A. Fuzzy Sets // Information and Control. — 1965. — № 8 .— P. 338—353.
58. Zuo Y. General notions of statistical depth functions / Y. Zuo, R. Serfling // Ann. Statist. — 2000. — Vol. 28, No. 2. — P. 461–482.
59. Mosler K. Multivariate dispersion, central regions and depth / K. Mosler. — New York. : Springer, 2002. — 291 p.
60. Environment using Oracle VM VirtualBox / S. Coter, G King.— California — 2016. — 30 с.
61. IT Service Management: An Introduction/ Jan Van Bon, G. Kemmerling, D. Pondman, Publisher: Van Haren Publishing, 2002, ISBN 90-806713-63
62. Bartolini C. Business-impact analysis and simulation of critical incidents in IT service management / C. Bartolini, C. Stefanelli, M. Tortonesi // 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009). — vol. 11. — no. 1. — June 2009. — 2009. — P. 9–16.
63. Introducing the IBM process reference model for IT. IBM Corporation. — 2008. — 21 p.
64. SLA Management Handbook. Vol. 1. Executive Overview. Ver. 2.1. TeleManagement Forum. — 2005. — 39 p.; Vol. 2. Concepts and Principles. Release 2.5. — TeleManagement Forum. — 2005. — 218 p.; Vol. 3. Service and Technology Examples. Ver. 2.1. — TeleManagement Forum. — 2005. — 89 p.
65. Bucu M.J., Chang R.N., Luan L.Z., Ward C., Wolf J.L., Yu P.S. Utility computing SLA management based upon business objectives // IBM Systems Journal.— 2004.— Vol.43.—№1.— P.159—178.
66. Boutaba R., Xiao J., Aib J. Cyber Planner: A comprehensive toolkit for network service providers. — 2008, IEEE. — P. 382—386.

67. Mahdavy J., Paxson V. IPPM metrics for measuring connectivity // RFC 2678/ Proposed Standard. — September. 1999. Bergnum K., Burgess M., Jonassen T.M., Fagernes S. On the stability of adaptive service management // eTransactions on Network and Service Management.—2006.— Vol.2.—№1.— P.13—21.
68. Russell J. OpenVZ / J. Russell, R. Cohn – Miami. –2012. – 144 с.
69. KVM. Linux Documentation [Електронний ресурс]. Доступ: <http://www.linux-kvm.org/page/Documents>.
70. Yan Chen. QoS requirements of Network Applications on the Internet/ Yan Chen, Toni Farley, Nong Ye // Information-Knowledge-System Management, January 2004. – С. 55–76.
71. Giovanni Giambene. Resource management in satellite networks // 2007. – С. 67–94.
72. Paxson V., Almes G., Mahdavy J., Mathis M. Framework for IP performance metrics //RFC 2330, Informational. — May. 1998.
73. Ролік О.І. Застосування агентського підходу до управління інформаційно-телекомунікаційною системою АСУ спеціального призначення / О.І. Ролік, П.Ф. Мажаровський, О.О. Покотило // Пріоритетні напрямки розвитку телекомунікаційних систем та мереж спеціального призначення: V наук.-практ. семін., 22 жовт. 2009 р.: доповіді та тези доповідей: — К.: ВІТІ НТУУ «КПІ», 2009. — С. 228—229.
74. Turnbull J. Pulling String with Puppet. / J. Turnbull. – New York – 2007. – 168 с.
75. Бесекерский В.А. Теория систем автоматического регулирования/ В.А. Бесекерский, Е.П. Попов. — М., «Наука». — 1975. — 768 с.
76. Теленик С.Ф. Система управління інформаційно-телекомунікаційною системою корпоративної АСУ / С.Ф. Теленик, О.І. Ролік, М.М. Букасов, Р.Л. Соколовський // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — 2006. — № 45.— С. 112—126.
77. Ролик А.И. Модель управления перераспределением ресурсов информационно-телекоммуникационной системы при изменении значимости

бизнес-процессов / А.И. Ролик// Автоматика. Автоматизация. Электротехнические комплексы и системы. ХГТУ, 2007.— №2 (20).— С. 73—82.

78. Теленик С.Ф. Управління доступом до обмежених ресурсів інформаційно-телекомунікаційної мережі АСУ військового призначення / Теленик С.Ф., Ролік О.І., Букасов М.М., Терещенко П.І // Сб. наук. праць ЦНДІ Збройних Сил України. — 2006. — №3 (37). — С. 33—43.

79. Теленик С.Ф. Моделі управління розподілом обмежених ресурсів в інформаційно-телекомунікаційній мережі АСУ / Теленик С.Ф., Ролік О.І., Букасов М.М. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: Єкотех, — 2006. — № 44.— С. 234—239.

80. Теленик С. Ф. Модель управління розподілом ресурсів інформаційно-телекомунікаційної системи збройних сил України / Теленик С. Ф., Ролік О.І., Терещенко П.І., Літвінцов О.В. // Збірник науков. праць ННДЦ оборонних технологій і воєнної безпеки України. — 2006. — №5 (34). — С. 117—124.

81. Ролик А.И. Анализ качества функционирования элементов информационно-телекоммуникационных систем / А.И. Ролик, Е.В. Глушко // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: «ВЕК+», 2008. — № 48. — С. 113—120.

82. Ролик А.И. Управление распределением ресурсов в программно конфигурируемых сетях / А.И. Ролик, Т.В. Кравченко, Н.В. Кравчун // Комп'ютерні системи і мережні технології (CSNT 2015): VIII Міжнар. наук.-техн. конф., 16-18 квітня 2015 р. м. Київ: зб. тез. – Київ, НАУ, 2015. – С. 56.

83. Кравченко Т.В. Управление ресурсами виртуальной ИТ-инфраструктуры / Т.В. Кравченко, Н.В. Кравчун // Праці студентської наукової конференції фізико-технічного факультету Донецького національного університету, за період 2014-2015 учбового року: зб. наук. праць. – Вінниця, ДонНУ, 2015. – с. 34.

84. Теленик С.Ф. Метод формування логічного висновку із залученням експертного комітету / С.Ф. Теленик, П.І. Бідюк, Л.О. Коршевніук, В.С. Хмелюк // Проблеми програмування. — 2008. — №4. — С. 73—83.

85. Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup", *Computers & Operations Research*, vol. 36, 2009, P. 3215-3223
86. E. E. Zachariadis, C. D. Tarantilis and C. T. Kiranoudis, "An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries", *European Journal of Operational Research*, vol. 202, P. 401-411
87. Коршевніюк Л.А., Бидюк П.И. Решение задачи распределения инвестиций на основе нечеткого логического вывода // Системні дослідження та інформаційні технології. — 2003. — № 2. — С. 34—42.
88. Sugeno M. An introductory survey of fuzzy control // *Information Sciences*. — No 36, 1985. — P. 59—83.
89. O. Shpur, B. Strykhalyuk, and O. Morushko, "The optimal distribution of optical resources between data centers for providing the required level of QoS," the 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2016 — P. 649–651.
90. C. Li, X. Liu, K. Sun, Y. Cao, F. Ma and B. Zhou, "A hybrid control strategy to support voltage in industrial active distribution networks", *IEEE Trans. Power Del.*, vol. 33, no. 6, 2018 — P. 2590-2602.
91. L. Gu, D Zeng, A Barnawi, S Guo, and I. Stojmenovic, "Optimal Task Placement with QoS Constraints in Geo-Distributed Data Centers Using DVFS," *IEEE Transactions on Computers*, vol. 64, issue. 7, 2015 — P. 2049-2059.
92. H. R. Baghaee, M. Mirsalim, G. B. Gharehpetian and H. A. Talebi, "Eigenvalue robustness and time delay analysis of hierarchical control scheme in multi-DER microgrid to enhance small/large-signal stability using complementary loop and fuzzy logic controller", *J. Circuits Syst. Comput.*, vol. 26, no. 6, Jun. 2017.
93. Коршевніюк Л.О. Система нечіткого логічного виводу із зваженою істинністю / Л.О. Коршевніюк, М.Ю. Мінін // Единое информационное пространство '2004: Сборник докладов II-й Международной научно-практической конференции. — Днепропетровск: ИПК ИнКомЦентра УГХТУ, 2004. — С. 114—117.

94. P. Moutis and N. D. Hatziargyriou, "Decision trees-aided active power reduction of a virtual power plant for power system over-frequency mitigation", IEEE Trans. Ind. Informat., vol. 11, no.1, 2015. — P. 251-261.

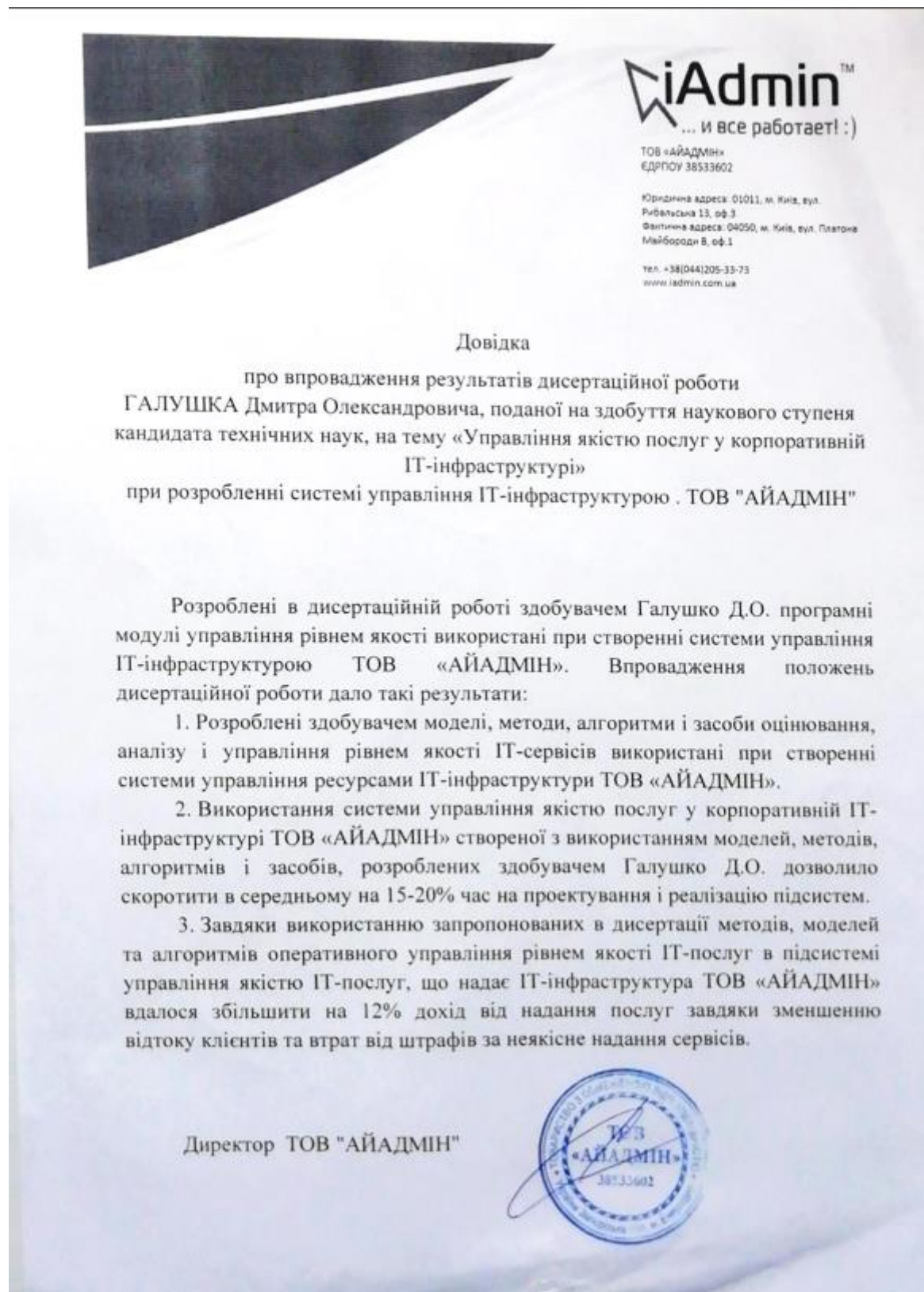
95. Mitsuo Gen. Recent advances in hybrid evolutionary algorithms for multiobjective manufacturing scheduling / Mitsuo Gen, Wenqiang Zhang, Lin Lin, YoungSu Yun // Elsevier Science / Computers & Industrial Engineering, 2017. — P. 616-633.

96. Теленик С.Ф. Забезпечення процесів діяльності ресурсами з визначеним рівнем надійності в інформаційно-телекомунікаційній системі спеціального призначення / Теленик С.Ф., Ролік О.І., Терещенко П.І., Букасов М.М. // Збірник наукових Військового інституту телекомунікацій та інформатизації національного технічного університету України «Київський політехнічний інститут». — Випуск № 3. — Київ: ВІТІ НТУУ «КПІ», 2006. — С. 134—138.

97. Ролик А.И. Управление устранением неисправностей в ИТ-системах / Ролик А.И., Тимофеева Ю.С., Турский Н.И. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: «БЕК+», — 2008. — № 49. — С. 94—107.

98. H. R. Baghaee, M. Mirsalim, G. B. Gharehpetian and H. A. Talebi, "Decentralized sliding mode control of wg/pv/fc microgrids under unbalanced and nonlinear load conditions for on-and off-grid modes", IEEE Syst. J., vol. 12, no.4, 2018. — P. 3108-3119

99. Брукс П. Метрики для управления ИТ-услугами / П. Брукс; Пер. с англ. — М.: Альпина Бизнес Букс, 2008. — 283 с.



Довідка

про впровадження результатів дисертаційної роботи
ГАЛУШКА Дмитра Олександровича, поданої на здобуття наукового ступеня
кандидата технічних наук, на тему «Управління якістю послуг у корпоративній
ІТ-інфраструктурі»
при розробленні системи управління ІТ-інфраструктурою ТОВ «СІТІУС ПРО»

Розроблені в дисертаційній роботі здобувачем Галушко Д.О. програмні модулі управління рівнем якості використані при створенні системи управління ІТ-інфраструктурою ТОВ «СІТІУС ПРО». Впровадження положень дисертаційної роботи дало такі результати:

1. Розроблені здобувачем програмні модулі управління рівнем якості використані при створенні автоматичної системи управління ІТ-інфраструктурою шляхом реалізації методів ефективного використання ресурсів ІТ-інфраструктури ТОВ «СІТІУС ПРО».


2. Розроблений структурний метод навчання нейромережових класифікаторів дозволив створити уніфікований модуль управління ІТ-інфраструктурою, призначений для підтримки узгодженого рівня якості ІТ-послуг, що надає ТОВ «СІТІУС ПРО». Це дозволило скоротити в середньому на 15-20% час на проектування і реалізацію підсистем.

3. Використання запропонованих в дисертаційній роботі моделей та методів перерозподілу ресурсів в підсистемі управління консолідованими інформаційно-обчислювальними ресурсами, що є складовою системи управління ІТ-інфраструктурою ТОВ «СІТІУС ПРО», дозволило скоротити витрати на експлуатацію серверного парку на 10-15% з одночасним збільшенням ефективності його використання при забезпеченні стабільного рівня якості надання користувачам інформаційно-телекомунікаційних послуг.

Комерційний Директор ТОВ «СІТІУС ПРО»



«ЗАТВЕРДЖУЮ»
Перший проректор
Національного технічного університету
України «Київський політехнічний інститут
імені Ігоря Сікорського»
академік НАН України

 Якименко Ю.І.
« 12 » січня 2021 р.

АКТ

про впровадження результатів кандидатської дисертаційної роботи
ГАЛУШКА Дмитра Олександровича
від 12 січня 2021 року

Комісія у складі декана факультету інформатики та обчислювальної техніки НТУУ «КПІ» С.Ф.Теленика і завідувача кафедри автоматики та управління в технічних системах О.І.Роліка склала цей акт про те, що результати наукових досліджень за темою кандидатської дисертаційної роботи Д.О.Галушка «Управління якістю послуг у корпоративній IT-інфраструктурі» використовуються в навчальному процесі кафедри автоматики та управління в технічних системах Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», починаючи з 2019-2020 навчального року в матеріалах лекцій з навчальних дисциплін «Компоненти програмної інженерії», а саме: «Компоненти програмної інженерії-1. Аналіз вимог до програмного забезпечення», «Компоненти програмної інженерії-2. Моделювання та аналіз програмного забезпечення», «Компоненти програмної інженерії-3. Якість програмного забезпечення та тестування»

На основі виконаних досліджень здобувачем розроблено методичне забезпечення, яке використане в навчальному процесі при викладанні дисциплін «Компоненти програмної інженерії-1. Аналіз вимог до програмного забезпечення», «Компоненти програмної інженерії-2. Моделювання та аналіз програмного забезпечення», «Компоненти програмної інженерії - 3. Якість програмного забезпечення та тестування».

Впровадженні наступні результати кандидатської дисертаційної роботи:

- моделі та метод оцінки якості ІТ-послуг;
- методи управління елементами ІТ-інфраструктури;
- методи управління якістю ІТ-послуг, що надаються ІТ-інфраструктурою.

Впровадження результатів дисертаційної роботи Галушка Д.О. дає можливість ознайомити студентів з сучасним поглядом на проектування систем управління інфраструктурою провайдера ІТ-послуг, визначення рівня якості ІТ-послуг та керування рівня якості ІТ-послуг шляхом перерозподілу ІТ-ресурсів провайдера ІТ-послуг; новими підходами до управління ресурсами ІТ-інфраструктури; підходами до розробки власних методологій та алгоритмів вирішення задач, що виникають при управлінні ІТ-інфраструктурами різного призначення та масштабу.

Декан ФІОТ д.т.н., проф.
Зав. каф. АУТС, д.т.н., проф.



Теленик С.Ф.
Ролік О.І.



Додаток Б – Перелік наукових публікацій за темою дисертації та відомості про апробацію результатів

1. Rolik O., Kolesnik V., Halushko D. Decomposition-Compensation Method for IT Service Management. In: Kulczycki P., Kóczy L., Mesiar R., Kacprzyk J. (eds) Information Technology and Computational Physics. CITCEP 2016. Advances in Intelligent Systems and Computing, vol 462, Springer, Cham, 2017, pp. 89-107.
2. Telenyk S. Models and methods of resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko // Technical transaction. Automatic control. – Politechnica Krakowska, 2013. – vol. 4-AC. – pp. 41–52
3. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // Information Technology, Computational and Experimental Physics. Kulczycki P., Kowalski P. A., Łukasik S. (eds). – AGN-UST. – 2016. – pp. 13-16.
4. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ»: Інформатика, управління та обчислювальна техніка. – К.: ВЕК+, 2010. – № 52. – С. 39–52
5. Ролик А.И. Оценка качества предоставления мультимедийных сервисов с использованием нейросетевого классификатора / Ролик А.И., Галушко Д.А., Барна В.В., Томащук А.В., Ясочка М.В. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Століття +, 2015. – № 63. – С. 25–30.
6. Ролик А.И. Метод нечеткой непараметрической оценки качества функционирования элементов информационно-телекоммуникационных систем / А.И. Ролик, Д.А. Галушко, П.Ф. Можаровский, А.А. Андриенко, В.М. Вовк // Адаптивні системи автоматичного управління. Міжвідомчий наук.-техн. зб. – Дніпропетровськ: Системні технології, 2012.– Вип. 19 (39).– С. 115–125.

7. Structural Optimization of Neural Network for Qualitative Evaluation Method of IT-Infrastructure Functioning / Yaroslav Yu. Dorogyi, Sergii F. Telenik, Dmytro A. Halushko, Vasyl V. Tsurkan // Information and telecommunication sciences : international research journal. – 2015. – Vol. 6, N. 2(11). – Pp. 36–43. – Bibliogr.: 15 ref

8. Теленик С.Ф. Система управления ИТ-инфраструктурой – путь к повышению эффективности функционирования предприятия / С.Ф. Теленик, А.И. Ролик, М.М. Букасов, А.В. Волошин, Д.А. Галушко // Інформаційні технології – інструмент підвищення конкурентоздатності підприємств: 9–10 груд. 2009.: матеріали конф. – К.: УкрНЦ РІТ, 2009. – С. 30–33.

9. Ролик А.И. Метод оценки состояния элементов информационно-телекоммуникационных систем на основе нечетких нейронных сетей/ А.И. Ролик, Д.А. Галушко, Ю.А. Кононенко // Обчислювальний інтелект (Результати, проблеми, перспективи): матеріали II-ї Міжнар. наук.-техн. конф. «Обчислювальний інтелект (QI-2013)», м. Черкаси, 14-18 травня 2013 р. – Черкаси: – 2013. – С. 233

10. Telenyk S. Data center resource management for VPS hosting / S. Telenyk, O. Rolik, M. Bukasov, D. Halushko// : матеріали конф Proc. of the International Congress on Control and Information Processing 2013 (ICCIP'13) 7–8 December, Cracow, Poland. – 2013.

11. Ролік О.І. Метод оцінки стану елементів інформаційно-телекомунікаційних систем на базі апарату нечіткої логіки / О.І. Ролік, Д.О. Галушко, І.М. Плехова, Б.В. Ступак // : матеріали конф Physical and technological problems of radio engineering devices, telecommunication, nano- and microelectronics. Proceedings of the 3th international scientific-practical conference. Chernivtsi, Ukraine, October 24–26, 2013. – Чернівці: КОД. – 2013. – С. 104–105.

12. Qualitative evaluation method of IT-infrastructure elements functioning / S. Telenyk, O. Rolick, M. Bukasov, Y. Dorogiy, D. Halushko, A. Pysarenko // : матеріали конф 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Odessa, 2014, pp. 165-169.

13. Ролик А.И. Метод оценки качества телекоммуникационных сервисов на основе нейронных сетей со структурным обучением / А.И. Ролик, Д.А. Галушко, Д.С. Захаров, А.В. Томащук// Обчислювальний інтелект (результати, проблеми, перспективи): III-я Міжнародна наук.-практ. конф., 12-15 травня 2015 р. Київ-Черкаси: матеріали. – Черкаси: видавець Чабаненко Ю., 2015. – С. 250—251.

14. Rolik O. Neural network approach for resource allocation in IT-infrastructure Management System / O. Rolik, V. Kolesnik, D. Halushko // : матеріали конф Proc. of the Congress on Information Technology, Computational and Experimental Physics 2013 (CITCEP'15) 18–20 December, Cracow, Poland. – 2015. – p. 176–179.

15. Ролик А.И. / А.И. Ролик, Д.А. Галушко, Н.В. Кравчун, Т.В. Кравченко // : матеріали конф ICACIT-2015: 3-я Міжнар. конф. з автоматичного управління та інформаційних технологій, 11-13 грудня 2015 р. м. Київ: матеріали. – К., 2015. – с. 80–83.

16. Rolik O. Decomposition-compensation method of service level management in corporate IT infrastructures with the use of adaptive genetic algorithm / O. Rolik, V. Kolesnik and D. Halushko// Proc. of the International Conference Radio Electronics & Info Communications (UkrMiCo). – Kiev, Ukraine. – 2016. – pp. 1–5.

17. D. Halushko, O. Rolik, and V. Samoty, “A Load Balancing Mechanism Based on Fuzzy Nonparametric Analysis of QoS Parameters” in Proc. CLOUD COMPUTING 2017: The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, February 19-23, 2017 Athens, Greece, IARIA, 2017, матеріали конф; pp. 102-107.